AGU PUBLICATIONS

Water Resources Research

RESEARCH ARTICLE

10.1002/2014WR016875

Key Points:

- Mutual information between system output and input was applied to determine system time lag vector
- Mutual information weighted average method was developed for time series embedding
- River water salinity was used to reconstruct ANN model input for simulation of pore water salinity

Correspondence to:
F. Zheng,
fzheng@sfwmd.gov

Citation:
Zheng, F., Y. Wan, K. Song, D. Sun, and M. Hedgepeth (2016), Reconstructing input for artificial neural networks based on embedding theory and mutual information to simulate soil pore water salinity in tidal floodplain, *Water Resour. Res.*, *52*, 511–532, doi:10.1002/2014WR016875.

Received 2 JAN 2015
Accepted 23 DEC 2015
Accepted article online 8 JAN 2016
Published online 29 JAN 2016

Reconstructing input for artificial neural networks based on embedding theory and mutual information to simulate soil pore water salinity in tidal floodplain

Fawen Zheng[1], Yongshan Wan[1], Keunyea Song[2], Detong Sun[1], and Marion Hedgepeth[1]

[1]Coastal Ecosystems Section, South Florida Water Management District, West Palm Beach, Florida, USA, [2]School of Natural Resources, University of Nebraska-Lincoln, Lincoln, Nebraska, USA

Abstract Soil pore water salinity plays an important role in the distribution of vegetation and biogeochemical processes in coastal floodplain ecosystems. In this study, artificial neural networks (ANNs) were applied to simulate the pore water salinity of a tidal floodplain in Florida. We present an approach based on embedding theory with mutual information to reconstruct ANN model input time series from one system state variable. Mutual information between system output and input was computed and the local minimum mutual information points were used to determine a time lag vector for time series embedding and reconstruction, with which the mutual information weighted average method was developed to compute the components of reconstructed time series. The optimal embedding dimension was obtained by optimizing model performance. The method was applied to simulate soil pore water salinity dynamics at 12 probe locations in the tidal floodplain influenced by saltwater intrusion using 4 years (2005–2008) data, in which adjacent river water salinity was used to reconstruct model input. The simulated electrical conductivity of the pore water showed close agreement with field observations ($RMSE \leq 0.031$ S/m and $R^2 \geq 0.897$), suggesting the reconstructed input by the proposed approach provided adequate input information for ANN modeling. Multiple linear regression model, partial mutual information algorithm for input variable selection, *k*-NN algorithm, and simple time delay embedding were also used to further verify the merit of the proposed approach.

1. Introduction

Intrusion of saltwater into coastal zones is increasingly observed, linked to rising sea level, lowered groundwater stage, and decreased freshwater inflow resulting from climate change and anthropogenic activities [*Barlow and Reichard*, 2010]. Saltwater intrusion usually leads to a substantial increase in soil pore water salinity in tidal floodplains, which eventually results in undesirable changes of vegetation species [*Neubauer*, 2013]. Thus, understanding and predicting the dynamics of soil pore water salinity in tidal floodplains or coastal wetlands are crucial to the management and restoration of degraded plant communities in these habitats.

In a tidal floodplain or coastal wetland, pore water salinity is a function of many variables, including tide stage, river water salinity, freshwater inflow from the upstream watershed, groundwater stage, temperature, rainfall, and evapotranspiration [*Kaplan et al*., 2010]. Usually only one or a few of these variables are monitored on a routine basis. Mechanistic process-based modeling of the pore water salinity requires a thorough understanding of system dynamics along with detailed mathematical representations of the various processes and a large amount of environmental data for model development [*Morris*, 1995]. Simulation of pore water salinity dynamics in a tidal floodplain using process-based model is difficult due to the complicated relationships among the physical, chemical, and biological processes, extensive data requirements, as well as complex boundary conditions. Artificial neural networks can provide an effective alternative to mechanistic modeling in such situations [*Patel et al*., 2002].

Artificial neural networks (ANNs) have been widely used to simulate complex and nonlinear relationships in water resources [*Maier and Dandy*, 2000; *Dawson and Wilby*, 2001; *Maier et al*., 2010]. ANNs belong to the class of data-driven approaches, in which the determination of input variables is extremely important [*Maier and Dandy*, 2000; *Bowden et al*., 2005a, 2005b; *May et al*., 2008]. Despite successful applications of ANNs reported in the literature, the determination of appropriate model input remains one of the most challenging tasks [*Maier and Dandy*, 1997, 2000; *Maier et al*., 2010]. The difficulty lies in two aspects. First, ANN model is still considered as a black box in the modeling community, and one of the main reasons to use ANNs is the incomplete understanding of the modeled system [*May et al*., 2008; *Zhao et al*., 2012]. This includes the incomplete knowledge of the complex underlying processes that govern the observed system dynamics, as well as the inadequate identification of significant system input variables that drive the system dynamics. Second, even if the system input variables are adequately identified with a priori knowledge and understanding of the system, data are not always available for all variables identified.

When a set of relevant system input variables is identified and data for the variables are available, the determination of inputs for ANN, or simply the input variables selection (IVS) involves selecting a subset of k variables, S, from an initial candidate set, C $\left(S^k \subset C\right)$, which comprises the set of all potential inputs candidates to the model [*May et al*., 2008]. A number of IVS techniques have been developed for ANNs [*May et al*., 2011], such as cross-correlation [*Huang and Foo*, 2002], partial mutual information [*Sharma*, 2000; *Sharma and Mehrotra*, 2014], self-organization maps (SOMs) [*Bowden et al*., 2005a, 2005b], stepwise selection [*Maier et al*., 1998], and Gamma test [*Končar*, 1997; *Remesan et al*., 2008; *Elshorbagy et al*., 2010a, 2010b]. However, little attention has been given to developing methods for the determination of ANN input when system input variables are partially identified, meaning that set C does not contain all input candidates. Fox example, only one input variable is known with data available. A possible approach for such a situation is time series embedding if the system is chaotic. In time series embedding, a system state variable is considered as the projection of the whole system onto a coordinate. With an appropriate embedding method, the information of unobserved state variables can be encapsulated in a multiple time series reconstructed by the data of the observed state variable [*Casdagli et al*., 1991]. This reconstructed time series can be used as model input for ANNs. For example, *Elshorbagy et al*. [2002] applied time delay uniform embedding method to reconstruct model input to estimate missing streamflow data using ANNs with available flow data. The input and output data in this case are the same variable (i.e., streamflow). Little attention is given to embedding techniques when input and output are different variables.

In this study, we developed a mutual information-based embedding approach to reconstruct input time series for ANN simulation of pore water salinity (output) in the tidal floodplain of the Loxahatchee River Estuary (South Florida) using measured river water salinity data (input). We first applied mutual information to determine the time lag vector representing statistical lag features between system output and input. Then, we developed a mutual information weighted average (MIWA) method to compute the components of the constructed time series in both nonuniform and uniform embedding. Using the reconstructed time series as model input, we trained ANNs and optimized model performance by searching for an optimal embedding dimension. This resulted in identification of the optimal input and the final ANN model. Finally, through model performance evaluation, we employed a few related modeling techniques including the time lagged multiple linear regression model, k-nearest neighbors model with MIWA embedding, ANN using partial mutual information for input selection, and ANN using simple time delay embedding to further verify the merit of the proposed approach.

2. Study Site

The Loxahatchee River is located on the southeastern coast of Florida, USA (Figure 1), consisting of the Southwest Fork, the North Fork, and the Northwest Fork with the longest tidal reach. Freshwater inflow from C-18 Canal to the Southwest Fork is controlled by Structure S46. The Northwest Fork receives freshwater mainly from the C-18 Canal through the G92 structure and Lainhart Dam. The three forks converge at approximately 3.2 river km (RK) upstream from Jupiter Inlet where the river empties into the Atlantic Ocean. Over the last century, the natural hydrology of the Loxahatchee River has been altered by drainage activities associated with agriculture, urbanization, and permanently opening the Jupiter Inlet for navigation [*Roberts et al*., 2008]. This resulted in saltwater encroachment along the river, which has led to a loss of riverine cypress forest in downstream portions of the river and upstream migration of mangroves [*Roberts et al*., 2006, 2008; *Wan et al*., 2015]. These shifts in vegetation have been directly linked to increased soil pore water salinity and hydroperiod changes within the floodplain [*SFWMD*, 2006; *Kaplan et al*., 2010].

10.1002/2014WR016875

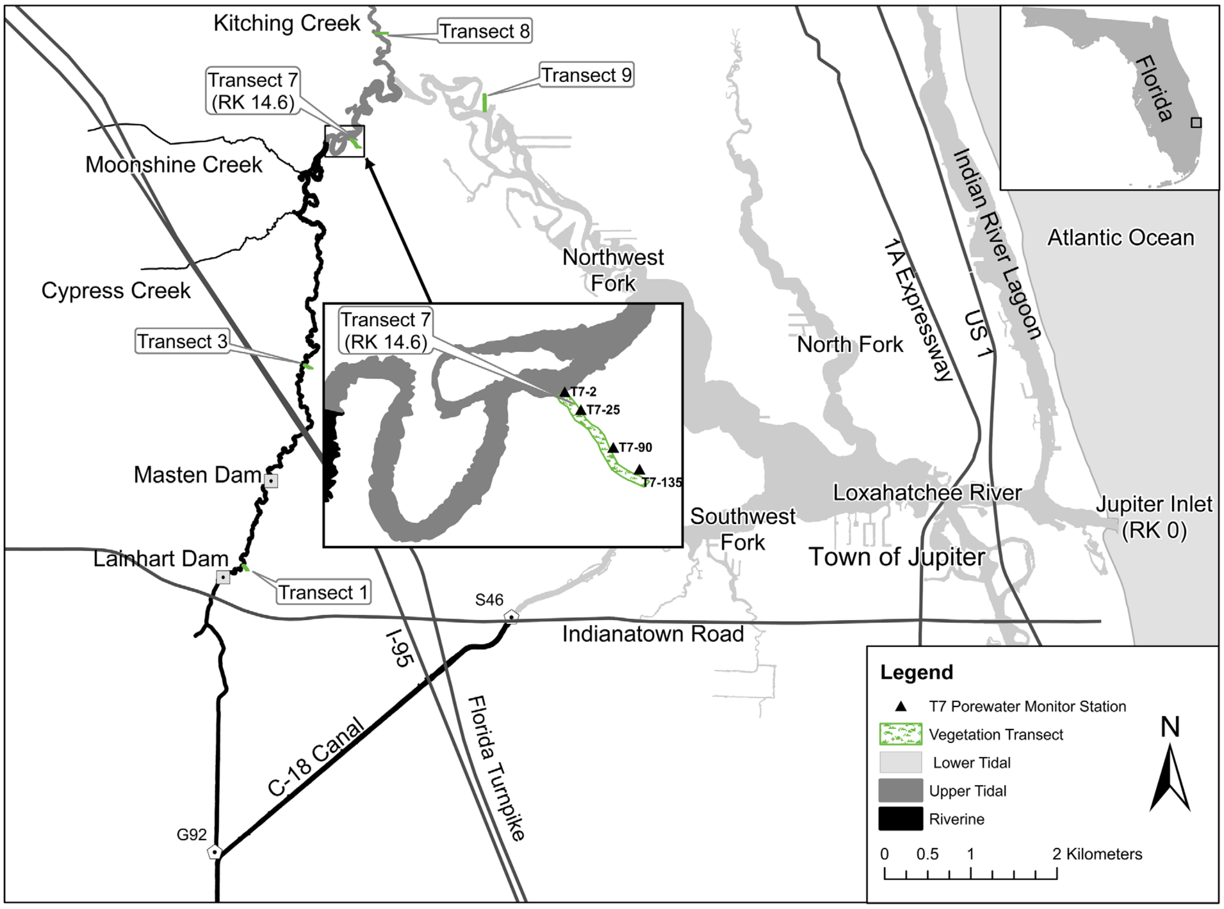

Figure 1. Map of study site in the Loxahatchee River, Florida.

3. Data Collection

To support the restoration and protection of the Loxahatchee River ecosystem, 10 sampling transects were established by the South Florida Water Management District (SFWMD) to investigate changes in vegetative coverage and species composition along the floodplain (the location of some transects are shown in Figure 1) [*SFWMD*, 2006]. River water salinity and groundwater stage/salinity were monitored at selected transects. In particular, soil moisture and pore water salinity were measured along Transects 7 (T7) from 2005 to 2008. T7 is located in the upper tidal floodplain reach of the Northwest Fork (Figure 1) and experiences significant variation in both river and floodplain pore water salinities [*Kaplan et al.*, 2010]. T7 begins at RK 14.6 in the river channel and runs southeast for 150 m through the floodplain to the edge of the uplands with elevations ranging from 3.07 m (NGVD29) in the upland to 0.40 m (NGVD29) in the floodplain. Four stations were established on T7 to monitor soil pore water characteristics (station nomenclature (T7-2, T7-25, T7-90, and T7-135) is based on the distance (m) from the riverbank edge, see Figure 1 insert and Figure 2). At each station, 24 frequency domain reflectometry dielectric probes (Hydra Probe: Steven Water Monitoring Systems, Beaverton, OR) were deployed at three different elevations (black squares in Figure 2) to measure soil moisture, soil temperature, and pore water salinity as electrical conductivity (EC, S/m) every 30 min. Each cluster of three probes was wired to a field data logger (CR10/CR10-X, Campbell Scientific, Logan, UT). Wells also were set up to monitor groundwater stage and salinity at each station. The probe location name denotes the station and its elevation, for example, T7-2(0.30) is the probe at elevation 0.30 m (NGVD29) at station T7-2. *Mortl et al.* [2011] and *Kaplan et al.* [2010] provided additional details about the experimental setup. River water salinity at RK 14.6 was measured every 15 min by United States Geological Survey (USGS). Due to harsh field conditions (e.g., hurricanes, frequent lightning strikes, etc.), there are gaps in both river water and pore water salinity data record. Daily data were used for modeling, and Table 1 lists the number of available data points used in this study.

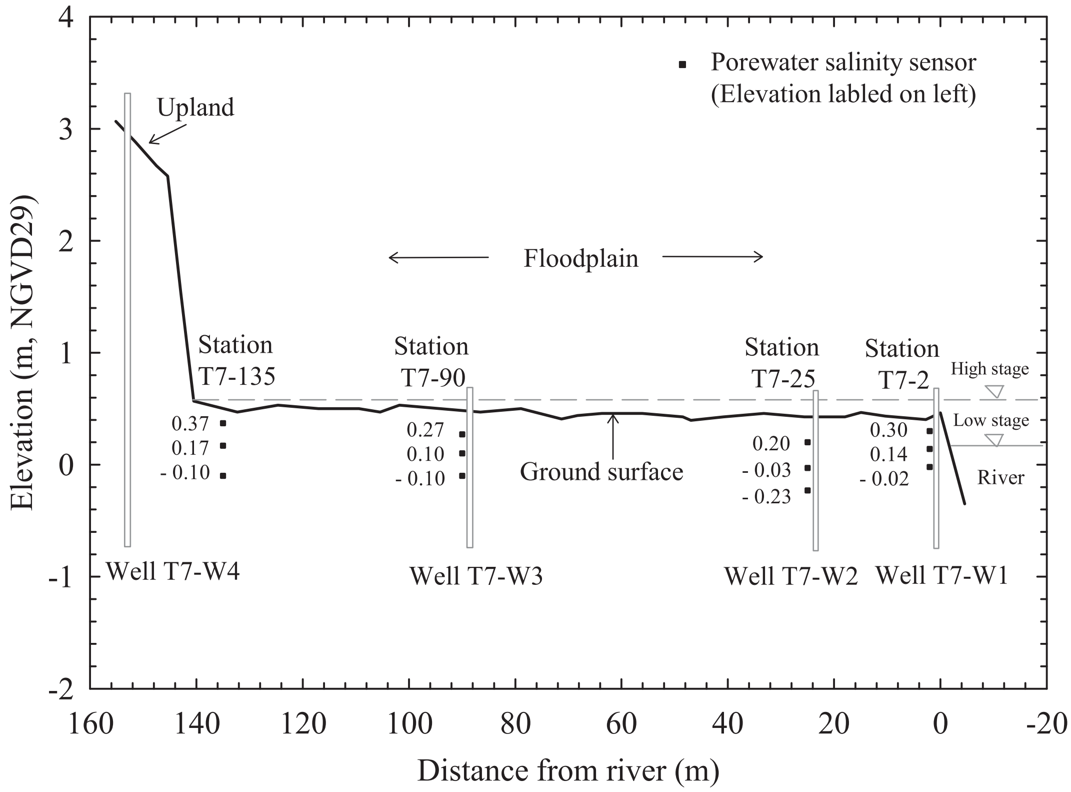

Figure 2. Transect 7 topographic profile, groundwater wells, and pore water probe locations.

4. Factors Influencing Pore Water Salinity in T7 River-Floodplain System

Previous studies [*SFWMD*, 2006; *Kaplan et al*., 2010; *SFWMD/LRD/FDEP*, 2012] indicated that the change in pore water salinity at T7 floodplain was influenced by a combination of factors, including the tidal cycle, river salinity at RK 14.6, river and groundwater stage, freshwater inflow from the upstream watershed, rainfall, evapotranspiration, and water temperature. The pore water salinity dynamics at a specific location along T7 are generally expressed as:

$$y(t)=f\left(x(t-i),\ \ h_{gw}(t-i), h_r(t-i), et(t-i), p(t-i),\ \ T(t-i)\right),\ \ i=0,1,2,\ldots \tag{1}$$

where, t is time, i is time lag, $y(t)$ is T7 floodplain pore water salinity, $x(t)$ is river water salinity at RK 14.6, $h_{gw}(t)$ is groundwater stage, $h_r(t)$ is river stage at RK 14.6, $et(t)$ is evapotranspiration, $p(t)$is rainfall,

Table 1. Data Availability and Embedding Parameters for Artificial Neural Network Models Developed for 12 Probe Locations at Transect T7 in Loxahatchee River

Station	Probe Location Name	Total Number of Data Points	Final Time Lag Vector	Embedding Method	Embedding Dimension	Embedding Window (days)
T7-2	T7-2(0.30)	1088	(89,193,297,369)	Nonuniform	4	369
	T7-2(0.14)	851	(16,97,174,299,380)	Nonuniform	5	380
	T7-2(−0.02)	1047	(51,127,211,301,348)	Nonuniform	5	348
T7-25	T7-25(0.20)	1076	(86,172,258,344)	Uniform	4	344
	T7-25(−0.03)	1029	(86,172,258,344)	Uniform	4	344
	T7-25(−0.23)	1111	(110,204,295,386)	Nonuniform	4	386
T7-90	T7-90(0.27)	1095	(20,76,126,168,213,303,364)	Nonuniform	7	364
	T7-90(0.10)	992	(22,84,168,251,327,379)	Nonuniform	6	379
	T7-90(−0.10)	1032	(74,192,249,310,365)	Nonuniform	5	365
T7-135	T7-135(0.37)	851	(18,70,136,214,339)	Nonuniform	5	339
	T7-135(0.17)	893	(72,108,150,214,339)	Nonuniform	5	339
	T7-135(−0.10)	304	(112,224,336)	Uniform	3	336

and $T(t)$ is water temperature [*Kaplan et al*., 2010]. When an ANN model is used to simulate f in equation (1), traditionally all independent variables on the right-hand side of equation (1) are required as input. Among these variables, river water salinity at RK 14.6, $x(t)$ was the direct source causing the variation of floodplain pore water salinity [*Kaplan et al*., 2010]. We speculated that one can apply embedding theory to reconstruct ANN model input by embedding $x(t)$ in simulation of $y(t)$ when the observed data of other variables in the equation are not available.

5. ANN Modeling Program Description

A multilayer feed forward ANN, known as Multi-Layer Perceptron (MLP), which can learn a relationship of arbitrary complexity between input and output variables [*Shu and Burn*, 2004], was used for simulation of pore water salinity in this study. The MLP consisted of one input layer, one hidden layer, and one output layer and was coded in FORTRAN 90/95. The error back propagation algorithm was employed to train MLP for weight adjustment with a momentum term added to avoid erroneous oscillation at a local minimum [*Hertz et al*., 1991]. A hyperbolic tangent transfer function was used for the neurons in the hidden layer as an activation function. The output layer used a linear transfer function since it has the benefit of potentially unbounded output to match the characteristics of the actual data [*Shu and Burn*, 2004]. The early stopping technique [*Bishop*, 1995] was applied to find the optimal epoch number in model training. We configured the model with only one node in the output layer. Once the input data were determined, the data were grouped into three different sets by randomly selecting without replacement: 70% of the data for model training, 15% for model testing, and 15% for model validation. The training data set was used to optimize model weights between layers while the testing data set was applied to determine the optimal epoch number using early stopping without involving in model weights optimization. Model weights were randomly initialized at the start of training, and the trained weights associated with the optimal epoch number were used in the final model validation with the validation data set. Input data were normalized using its minimum and maximum values but no transformation algorithms (e.g., logarithm transformation) were involved. A relatively small learning rate, $\eta=0.01$, and momentum parameter, $\alpha=0.55$, were used in model training. Models were evaluated based on their root-mean-square error ($RMSE$) and coefficient of determination (R^2). The optimal number of neuron nodes in the hidden layer was determined by trial-and-error.

6. Time Series Embedding and Reconstruction

Time series embedding and reconstruction, aka state (or phase) space reconstruction, is a technique developed from nonlinear time series analysis, which is an effective approach for characterizing system dynamics based on a single scalar time series. The concept was originated by *Whitney* [1936], numerically demonstrated by *Packard et al*. [1980], and theoretically proved by *Takens* [1981], which indicated that d-dimensional state space could be reconstructed by an m-dimensional time delay vector embedded from a single scalar time series as long as m is large enough.

Let $f: M \rightarrow M,\ M \in \mathbb{R}^d$, be a time evolution operator of a smooth dynamic system s on state space M (d-dimensional manifold), then

$$s(t+1)=f(s(t)) \tag{2}$$

If we have measurement function $h: M \rightarrow \mathbb{R}^D,\ D<d$ (often implicitly assuming $D=1$), then a D-dimension time series $x(t)$ is related to the dynamic system s by

$$x(t)=h(s(t))+\varepsilon(t) \tag{3}$$

where $h(\cdot)$ is measurement function, and $\varepsilon(t)$ is independent noise usually assumed as a random variable with Gaussian distribution $N(0,\sigma^2)$. For the case $D=1$, we can reconstruct a m-dimension $v(t) \in \mathbb{R}^m$ using the time lag τ found in $x(t)$:

$$\begin{aligned} v(t)&=\{v_0(t),\ v_1(t)\dots\ v_{k-1}(t),\ v_k(t),\ \dots,\ v_{m-1}(t)\} \\ &=\{x(t),\ x(t-\tau),\dots,\ x(t-(k-1)\tau), x(t-k\tau),\dots,\ x(t-(m-1)\tau)\} \end{aligned} \tag{4}$$

10.1002/2014WR016875

Let F be time evolution operator of the smooth dynamics of which state is defined by $v(t)$, i.e.,

$$v(t+1)=F(v(t)) \tag{5}$$

Takens [1981] proved that f and F are topographically equivalent when $m \geq 2d+1$. In other words, there exists a diffeomorphic (one to one differentiable with a one to one differentiable inverse) map $\phi : f \rightarrow F$, allowing us to estimate $F(v(t))=\phi \circ f \circ \phi^{-1}(v(t))$ [*Casdagli et al.*, 1991]. The map ϕ is also called the reconstruction function. Since F has all important dynamics aspects of f, we can use F for any purpose such as prediction, quantitative or statistical system characterizations, and computation of system dimensionality [*Judd and Mees*, 1998].

Since a uniform time lag τ is used in equation (4), it is called uniform embedding or regular embedding. For classic chaotic systems, such as Rössler system [*Rössler*, 1976] and Lorenz system [*Lorenz*, 1963], having a single and dominant periodicity or recurrence time, uniform embedding is the most effective and suitable. However, it may fail if a time series holds strong periodicities with greatly differing time scales [*Judd and Mees*, 1998]. This multiple time scale problem can be overcome with nonuniform embedding, aka irregular embedding, which uses nonuniform time lags [*Judd and Mees*, 1998]. In this case, equation (4) becomes

$$\begin{aligned} v(t) &= \{v_0(t), v_1(t)\ \ldots,\ v_k(t),\ \ldots,\ v_{m-1}(t)\} \\ &= \{x(t),\ x(t-\tau_1),\ \ldots, x(t-\tau_{k-1}), x(t-\tau_k), \ldots,\ x(t-\tau_{m-1})\} \end{aligned} \tag{6}$$

$\{\tau_1, \tau_2,\ \ldots,\ \tau_{m-1}\},\ 0<\tau_1<\tau_2<\ldots<\tau_{m-1}$, is called time lag vector. τ_{m-1} is embedding window. Note $\tau_0=0$, not included in the vector.

With equation (4) or (6), multivariate time series (vector time series) $\{v_0(t),\ v_1(t),\ \ldots, v_k(t),\ \ldots,\ v_{m-1}(t)\}$ are reconstructed by simple time delay (STD) from one single scalar time series $x(t)$. For uniform embedding, the kth component of $v(t)$ in equation (4) is

$$v_k(t)=h(s(t-k\tau))+\varepsilon(t-k\tau)=x(t-k\tau) \tag{7}$$

For nonuniform embedding, the kth component of $v(t)$ in equation (6) is

$$v_k(t)=h(s(t-\tau_k))+\varepsilon(t-\tau_k)=x(t-\tau_k) \tag{8}$$

A more general form to reconstruct $v_k(t)$ can be written as [*Fraser*, 1989]:

$$v_k(t)=\sum_i a_{k,i}x(t-i) \tag{9}$$

State space reconstruction holds the idea that the past and future of a time series contain information about unobserved state variables, which can be encapsulated in the delay vector defined by a format like equation (4) or (6) [*Casdagli et al.*, 1991]. Based on this idea and a comparison of equations (5) and (1), we conjecture that we can reconstruct a vector time series $v(t)$ with an appropriate embedding method like equation (4) or (6) using $x(t)$, one of independent variables of equation (1), so that the information of other independent variables in equation (1) is encapsulated in $v(t)$. Then, $v(t)$ can be used as ANN model input to simulate $y(t)$ in equation (1). Thus, equation (1) becomes $y(t+1)=f_{ANN}(v(t))$, which is similar to equation (5). However, our modeling tests indicated that embedding with equations (4) and (6) did not result in satisfactory simulation of pore water salinity most likely because the output $y(t)$ and components of the input $x(t)$ are not the same variable (note that equation (5) has the same variable in output and input). For that, we developed a new embedding method to compute components of $v(t)$ (see section 9).

For an infinite amount of noise-free data, the time lag τ can in principle be chosen arbitrarily according to *Takens*' embedding theorem. Unfortunately, with finite noise contaminated data, the theorem provides no guidelines for time delay embedding [*Fraser*, 1989]. Since *Takens* [1981] and *Packard et al.* [1980], many researchers have proposed practical approaches to determine time lag, including the first zero of the autocorrelation function [*Albano et al.*, 1987], average displacement [*Rosenstein et al.*, 1994], singular-value decomposition [*Albano et al.*, 1988], and the first mutual information [*Fraser and Swinney*, 1986]. The embedding dimension m can be determined by computing the dynamic invariants, such as correlation dimension and Lyapunov exponents, through increasing m until the dynamic invariants are stable. The method of false nearest neighbors [*Kennel et al.*, 1992] and its various extensions [*Cao*, 1997] are other alternatives for calculating m. This method applies a topological reasoning by increasing m until the geometry of time series does not change. In this study, mutual information was employed to identify system time lags and embedding dimension was determined by optimizing model performance.

7. Mutual Information

In contrast to correlation coefficient, which only measures linear dependence between two variables, mutual information can capture both linear and nonlinear dependence [*Fraser and Swinney*, 1986; *Cellucci et al*., 2005]. Mutual information can be derived from Shannon entropy [*Shannon*, 1948; *Cover and Thomas*, 1991].

Let bivariate discrete random variables x and y have time series as $x=\{x(t_1), x(t_2), \ldots, x(t_n)\}$ and $y=\{y(t_1), y(t_2), \ \ldots, y(t_n)\}$ with sample size of n. Dividing the ranges of $x_{\min} \sim x_{\max}$ and $y_{\min} \sim y_{\max}$ into N_x and N_y elements, respectively, the associated distributions of x and y in elements i $(=1, 2, \ldots, N_x)$ and j $(=1, 2, \ldots, N_y)$, $P_x(x_i)$ and $P_y(y_j)$ are determined by histogram of the elements. Let $O_{xy}(i,j)$ denote the occupancy of the (i,j) element of the partition of the x-y plane extending from $x_{\min}$ to $x_{\max}$ on x axis (with N_x elements) and from $y_{\min}$ to $y_{\max}$ on y axis (with N_y elements), the joint distribution of x and y in the element, $P_{xy}(x_i, y_j)=O_{xy}(i,j)/n$ and it has N_xN_y values. The average mutual information $I(x,y)$ between x and y over all elements is [*Cover and Thomas*, 1991; *Cellucci et al*., 2005]:

$$I(x,y)=\sum_{i=1}^{N_x}\sum_{j=1}^{N_y} P_{xy}(x_i, y_j) log_b\left[\frac{P_{xy}(x_i, y_j)}{(P_x(x_i)P_y(y_j))}\right] \tag{10}$$

It is common to have $N_x=N_y$, but not necessary. Mutual information $I(x,y)$ represents the reduction in the uncertainty of y due to the knowledge of x. Unit of $I(x,y)$ is bits for $b=2$.

The mutual information function has been applied to identify optimal time lags in state space reconstruction [*Fraser and Swinney*, 1986; *Rosenstein et al*., 1994]. In this study, it was employed to determine the time lags between system output and input due to nonlinearity. If we have system output y with time lag τ with respect to system input x, then mutual information becomes a function of τ

$$I(x(t), y(t+\tau))=\sum_{i=1}^{N_x}\sum_{j=1}^{N_y} P_{xy}(x_i(t), y_j(t+\tau)) log_b\left[\frac{P_{xy}(x_i(t), y_j(t+\tau))}{(P_x(x_i(t))P_y(y_j(t+\tau)))}\right] \tag{11}$$

Kernel density estimation techniques [*Moon et al*., 1995] and fixed bin histogram [*Fraser and Swinney*, 1986; *Cellucci et al*., 2005] are two commonly used approaches to quantify mutual information. In this study, the *Fraser*'s algorithm [*Fraser* and *Swinney*, 1986], using the equal probability partition method, was implemented to calculate mutual information. The algorithm is based on the invariance of the mutual information with respect to transformations acting on the individual coordinates and on a recursive sequence of partitions of the variable.

Let $x(t)$ denote river water salinity at RK 14.6 and $y(t+\tau)$ denote T7 soil pore water salinity with time lag τ, mutual information $I(x(t), y(t+\tau))$ $(\tau=0$–$500)$ were computed based on observed data for all 12 probe locations (Figure 3, showing only $\tau=0$–400). Due to the presence of noise in observed data, i.e., $\varepsilon(t)$ in equation (3), the computed mutual information variation with time lag is usually not smooth, making it difficult to locate the extrema. Filters are usually required to smooth mutual information [*Martinerie et al*., 1992]. Mutual information shown in Figure 3 was the result smoothed with a double pass five-point moving-average method.

The mutual information at 9 of the 12 probe locations exhibited irregular periodicity or quasi-periodicity. These nine locations included all three at station T7-2 (i.e., T7-2(0.30), T7-2(0.14), and T7-2(−0.02)), all three at T7-90 (i.e., T7-90(0.27), T7-90(0.10), and T7-90(−0.10)), one at T7-25 (i.e., T-25(−0.23)), and two at T7-135 (i.e., T7-135(0.37) and T7-135(0.17)). The local minimum mutual information points were marked with black circle points with the corresponding time lag $(\tau_1, \tau_2, \ \tau_3, \tau_4, \ldots)$. The remaining three probe locations showed little quasi-periodicity (T7-25(0.20) and T7-25(−0.03)) or no quasi-periodicity (T7-135(−0.10)) due to unknown reasons, part of which could be the presence of noise in the data. We were able to identify the first minimum mutual information point for the three probe locations marked by black circle point with τ_1 in Figure 3. The three stations closer to the river (T7-2, T7-25, and T7-90) had mutual information greater

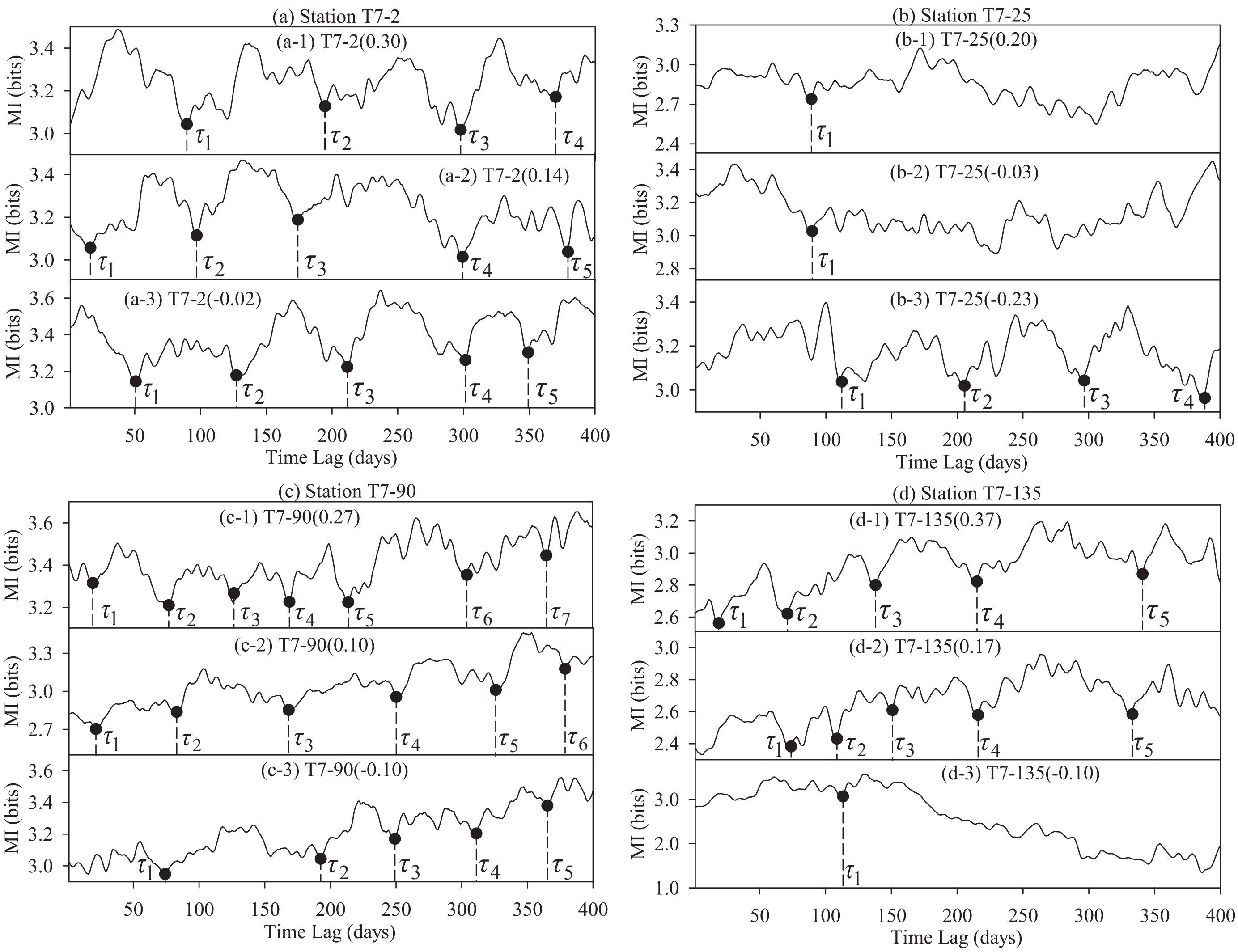

Figure 3. Plots of mutual information between river water salinity and pore water salinity at all 12 probe locations versus time lag (τ_k, $k=1, 2, \ldots$, are the time lags associated with local minimum mutual information. Mutual information exhibits irregular periodicity except in Figures 3b-1, 3b-2, and 3d-3).

than 2.6 bits, suggesting relatively high dependence of pore water salinity on river water salinity. Station T7-135, farthest from the river, had slightly lower mutual information, probably because this station was likely more impacted by groundwater from upland and less dependent on river water salinity than other stations [*Kaplan et al*., 2010]. Nevertheless, all probe locations at this station still had mutual information greater than around 2.4 bits except T7-135($-$0.10), which had relatively low mutual information ($<$2 bits when the time lag was greater than approximately 200 days). This was partly due to the limited amount of data at this probe location (only 304 data points were available), and using a large time lag significantly shortened the length of the data record for mutual information computation.

8. Chaotic Features in River Water Salinity

We aim to reconstruct ANNs input by embedding river water salinity at RK 14.6 for the simulation of pore water salinity at T7. This reconstruction is valid only when river water salinity at RK 14.6 is a chaotic time series. Here the chaotic features were explored using correlation dimension and maximum Lyapunov exponent based on observed daily data from 7 May 2007 to 25 Jan 2015.

8.1. Correlation Dimension

For m-dimension phase-space, the correlation integral $C(r)$ [*Grassberger and Procaccia*, 1983] is

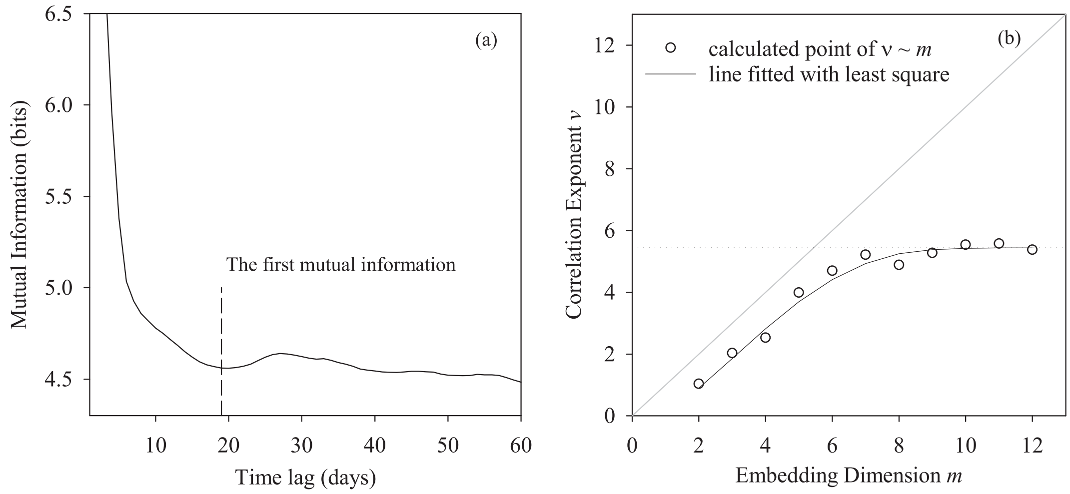

Figure 4. (a) Mutual information of river water salinity at RK 14.6 with time lag (the first local mutual information is at time lag of 19); (b) correlation exponent changes with embedding dimension and saturates at ∼5.34.

$$C(r)=\lim_{N\to\infty}\frac{2}{N(N-1)}\sum_{\substack{i,j\\0\le i<j\le n}}\Theta\left(r-||\boldsymbol{Y}_i-\boldsymbol{Y}_j||\right) \tag{12}$$

where N is the number of data points, $||\boldsymbol{Y}_i-\boldsymbol{Y}_j||$ is the Euclidean distance between points $\boldsymbol{Y}_i$ and $\boldsymbol{Y}_j$, r is the radius of the sphere centered on $\boldsymbol{Y}_i$ or $\boldsymbol{Y}_j$, and the Heaviside function $\Theta(x)$ is defined as $\Theta(x)=1$ for $x>0$, $\Theta(x)=0$ for $x\le 0$.

If the time series is characterized by an attractor, then the $C(r)$ and r are related according to

$$\underset{\substack{r\to 0\\N\to\infty}}{C(r)}\propto \alpha r^{v} \tag{13}$$

where α is a constant, and v is the correlation exponent and is the slope of $log\,C(r)$ versus $log\,r$ plot computed as

$$v=\lim_{\substack{r\to 0\\N\to\infty}}\frac{\log C(r)}{\log r} \tag{14}$$

The system is deemed to exhibit chaos if the correlation exponent v saturates with an increase in the embedding dimension. Otherwise, it is considered stochastic. The saturation value of v is defined as the correlation dimension (D_2). A total of 2821 daily data (7 May 2007 to 25 Jan 2015) of river water salinity at RK 14.6 were used for correlation dimension computation. The time lag of 19 days based on the first mutual information (Figure 4a) was used for embedding in the computation. Figure 4b is the result of $v\sim m$ and it shows correlation exponent v saturates at around 5.34, i.e., $D_2=5.34$.

8.2. Lyapunov Exponent

The Lyapunov exponents measure the average rate of divergence or convergence of orbits starting from nearby initial points in a dynamical system. A positive Lyapunov exponent is a fundamental indicator for the presence of chaotic attractors. In a state space, if $\boldsymbol{x}(0)$ and $\boldsymbol{x}_\epsilon(0)$ represent two initial points with distance ϵ, i.e., $||\boldsymbol{x}(0)-\boldsymbol{x}_\epsilon(0)||=\epsilon\le\epsilon_{max}$, after time t, they move to point $\boldsymbol{x}(t)$ and $\boldsymbol{x}_\epsilon(t)$ with distance $\delta(t)=||\boldsymbol{x}(t)-\boldsymbol{x}_\epsilon(t)||$. Then, we have $\delta(t)=\epsilon e^{\lambda t}$. Here λ is Lyapunov exponent and ϵ_{max} is the diameter used for searching nearest neighbors $\boldsymbol{x}_\epsilon(0)$. Each dimension has one Lyapunov exponent in a state space. Positive λ indicates that the system is chaotic. The maximal Lyanpunov exponent is defined as [*Kantz*, 1994]:

$$\lambda_{max}=\lim_{t\to\infty}\lim_{\epsilon\to 0}\frac{1}{t}\ln\left(\frac{||\boldsymbol{x}(t)-\boldsymbol{x}_\epsilon(t)||}{\epsilon}\right) \tag{15}$$

for almost all difference vectors $\mathbf{x}(0)-\mathbf{x}_\epsilon(0)$. The algorithm of *Kantz and Schreiber* [1997] (TISEAN software package [*Hegger and Kantz*, 1998]) was applied to compute maximum Lyapunov exponent for the river water salinity time series in this study, and positive $\lambda_{max}=0.0067$ was found. Theiler window of 57 days (three times of system time lag [*Heath*, 2000]), $\epsilon_{max}=0.01040$, 0.01849, and 0.03289, and 150 iterations were used in the computation.

Both correlation dimension and maximum Lyapunov exponent show that river water salinity at RK 14.6 is chaotic. Thus, embedding is applicable to state space reconstruction in the studied system.

9. Embedding Using Mutual Information Weighted Average

When equation (4) is applied to reconstruct vector time series $v(t)$, *Fraser and Swinney* [1986] suggested that the time lag is determined by the first minimum mutual information, i.e., the τ associated with the first minimum $I(x(t), x(t+\tau))$. The time lag chosen in this way is supposed to optimize the spread of the embedded time series without confusing the dynamics in the state space and without self-intersection [*Judd and Mees*, 1998]. This time lag reflects the recurrence within $x(t)$. According to equation (5), the $v(t)$ can be used as ANNs input to estimate unknown $x(t+1)$ [*Elshorbagy et al.*, 2002]. In this case, the model output is the same variable as the one used for input reconstruction, i.e., $x(t+1)=f_{ANN}(v(t))=f_{ANN}(x(t), x(t-\tau), \ldots, x(t-(m-1)\tau)$, where both input and output are the same variable x. When applying this concept to the case in which input and output are not the same variable, i.e., $y(t+1)=f_{ANN}(x(t), x(t-\tau), \ldots, x(t-(m-1)\tau))$ (y can be another system state variable), we are particularly interested in the time lag τ between $y(t)$ and $x(t)$, not the one within $x(t)$. In addition, a different embedding method from equation (4) may be needed accordingly. *Fraser and Swinney* [1986] used the Rössler chaotic system [*Rössler*, 1976] to demonstrate why the first mutual information was adopted as the criteria for selection of system time lag for simple time delay (STD) embedding in equation (4). Here we apply the same Rössler system data to show the concept of a proposed embedding method. The Rössler chaotic system is defined as:

$$\begin{cases} \dot{x}(t)=-(y(t)+z(t)) \\ \dot{y}(t)=x(t)+ay(t) \\ \dot{z}(t)=b+z(t)(x(t)-c) \end{cases} \tag{16}$$

with $a=0.15$, $b=0.20$, and $c=10.0$ [*Fraser and Swinney*, 1986; *Lall et al.*, 1996]. The fourth-order Runge-Kutta method was applied to solve the ordinary differential equations (ODEs) (equation (16)) with a fixed time step of $\pi/100$. A total of 65536 ($=2^{16}$) points were generated starting from an initial point (10, 0, 0). The mutual information between $x(t)$ and $y(t+\tau)$ ($\tau=1$–1000) was computed and shown in Figure 5 (only shown $\tau=1$–550). Mutual information represents the dependence between variables. Thus intuitively, from a modeling perspective, the $x(t)$ on which $y(t)$ has maximum dependence should be taken as components of $v(t)$ for model input. This means we should take the time lag ($\tau=53$) associated with the first maximum, instead of the first minimum, mutual information for uniform embedding by equation (4) as $y(t)$ has maximum dependence on $x(t-53)$. Using this time lag, considering embedding of five components in equation (4), we can obtain $v(t)=\{x(t),\ x(t-\tau), x(t-2\tau), x(t-3\tau),\ x(t-4\tau)\}$ ($\tau=53$). Components of $v(t)$ are associated with the mutual information points with multiple times of the time lag ($k\tau$, $\tau=53$, $k=1, 2, \ldots$) marked with open black circles in Figure 5. Note that the open black circles do not stick on local maximum mutual information point with increase of $k(\geq 2)$. They become non-characteristic (neither minimum nor maximum) and arbitrary. Arbitrarily selecting time lag $k\tau$ like this for uniform embedding by equation

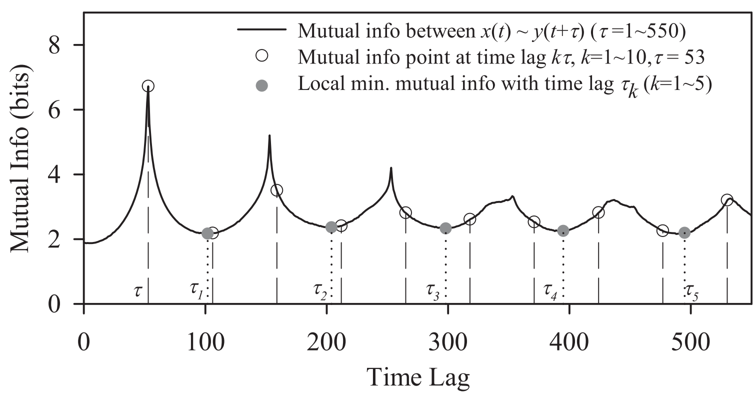

Figure 5. Mutual information between $x(\tau)$ and $y(t+\tau)$ (time lag $\tau=0$–550) of Rössler system. The first maximum mutual information is at $\tau=53$.

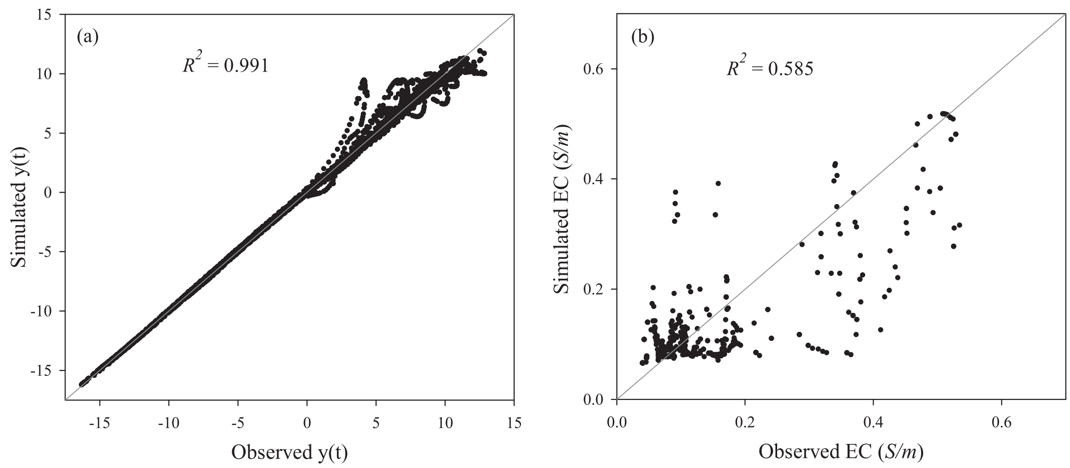

Figure 6. Scatterplot of observed versus simulated data using input reconstructed with time lags corresponding to the maximum mutual information points: (a) simulated and observed $y(t)$ of Rössler system; (b) simulated and observed electrical conductivity (EC) at probe location T7-2(0.30).

(4) does not make sense to modeling. Logically, the better way is to select the time lags associated with the local maximum mutual information in Figure 5, and apply nonuniform embedding equation (6) to reconstruct $v(t)$ as model input. With the $v(t)$ reconstructed under this consideration as model input ($m=2d+1 \approx 5$, $d \approx 2.01$ [*Konishi et al*., 1993] was used for embedding), we were able to well simulate $y(t)$ using the MLP program introduced in section 5. Similarly, we simulated pore water salinity of T7-2(0.30) but without satisfaction. Figure 6a is the scatterplot of simulated $y(t)$ versus observed $y(t)$ (i.e., the result by Runge-Kutta method) of the Rössler system. Figure 6b is the scatterplot of simulated pore water salinity at probe sensor location T7-2(0.30) versus observed data, for which the nonuniform time lag vector is {36, 137, 250, 327} selected based on the local maximum information in Figure 3a-1.

The Rössler system has relatively simple phase structure strictly governed by ODEs (equation (16)) with a fractal dimension of 2.01 [*Konishi et al*., 1993]. There is only one single nonlinear term (of second order), $z(t)x(t)$, in the equation for $\dot{z}(t)$. In contrast, the salinity dynamics in tidal floodplain like T7 is much more complicated with many more degrees of freedom. Mathematical representation of the salinity dynamics requires coupling of the Richard's equation (RE) to simulate water movement in vadose zone and convection-dispersion-adsorption equation (CDAE) to simulate salt transport in soil [*Simunek et al*., 1995; *Clemente et al*., 1997; *Buyuktas and Wallender*, 2002]. Both RE and CDAE are complex nonlinear partial differential equation. In addition, the salinity data observed at T7 have unavoidable noise but data of the Rössler system obtained by Runge-Kutta method are supposed to be with minimal noise. These differences explained the unacceptable simulation results of pore water salinity at T7-2(0.30) with the method that works for the Rössler system. It is common that a method works in a theoretic system but may not work well in a real-world system. The reason is that the components in the reconstructed input $v(t)$ using the maximum mutual information points do not contain sufficient state information and are not representative enough. If we look at the reconstructed input in equation (6), the data associated with mutual information between the time lag τ_{k-1} and τ_k are not included (when $\tau_k - \tau_{k-1} > 1$), i.e., the data points $x(t-i)(i = \tau_{k-1}+1, \ldots, \tau_k - 1)$ are left out. Those data may need to be taken into account in reconstructing representative input time series. Through tremendous modeling tests, we formulated an effective algorithm with consideration of those data points in reconstruction. This algorithm is described as the following.

For nonuniformed embedding, time lags are selected from the local minimum mutual information, for example, $\tau_1, \tau_2, \ldots, \tau_5$ in Figure 5. Then the data points between two time lags are averaged using the mutual information associated with them as weights since it quantifies the dependence between the two variables. We named it "mutual information weighted average" (MIWA). Generally, with time lag vector $\{\tau_1, \tau_2, \ldots, \tau_{k-1}, \tau_k, \ldots, \tau_m\}$ selected by local minimum mutual information, the kth component of new time series $\{v_1(t)\ldots,\ v_k(t), \ldots, v_m(t)\}$ is computed as

$$v_k(t)=\sum_{i=\tau_{k-1}+1}^{\tau_k}\left(\frac{I_i x(t-i)}{\sum_{i=\tau_{k-1}+1}^{\tau_k} I_i}\right),\quad k=1,2,\ldots,m \tag{17}$$

where I_i is the mutual information between output $y(t)$ and input $x(t-i)$, defined as

$I_i=I(y(t),x(t-i)),\ i=\tau_{k-1}+1,\ \tau_{k-1}+2,\ \ldots,\ \tau_k$. Let $a_{k,i}=\frac{I_i}{\sum_{i=\tau_{k-1}+1}^{\tau_k} I_i}$, equation (17) has exactly the same form as equation (9) generalized by *Fraser* [1989]. Using equation (17), we rewrite equation (6) in detail as:

$$\begin{aligned} v(t)&=\{v_1(\tau_1,t),v_2(\tau_2,t),\ldots,v_k(\tau_k,t),\ldots,v_m(\tau_m,t)\}\\ &=\left\{\sum_{i=1}^{\tau_1}\left(\frac{I_i x(t-i)}{\sum_{i=1}^{\tau_1} I_i}\right),\sum_{i=\tau_1+1}^{\tau_2}\left(\frac{I_i x(t-i)}{\sum_{i=\tau_1+1}^{\tau_2} I_i}\right),\ \ldots,\sum_{i=\tau_{k-1}+1}^{\tau_k}\left(\frac{I_i x(t-i)}{\sum_{i=\tau_{k-1}+1}^{\tau_k} I_i}\right),\ldots,\sum_{i=\tau_{m-1}+1}^{\tau_m}\left(\frac{I_i x(t-i)}{\sum_{i=\tau_{m-1}+1}^{\tau_m} I_i}\right)\right\} \end{aligned} \tag{18}$$

For uniform embedding, the uniform time lag is selected by the first minimum mutual information between $x(t)$ and $y(t+\tau)$. The kth component of new time series $\{v_1(t)\ldots,\ v_k(t),\ldots,v_m(t)\}$ is computed as:

$$v_k(t)=\sum_{i=(k-1)\tau+1}^{k\tau}\left(\frac{I_i x(t-i)}{\sum_{i=(k-1)\tau+1}^{k\tau} I_i}\right),\quad k=1,2,\ldots,m \tag{19}$$

where I_i is the mutual information between output $y(t)$ and input $x(t-i)$, defined as

$I_i=I(y(t),x(t-i)),\ i=\ (k-1)\tau+1,\ (k-1)\tau+2,\ldots,\ k\tau$. Using equation (19), we rewrite equation (4) in detail as:

$$\begin{aligned} v(t)&=\{v_1(\tau,t),v_2(2\tau,t),\ldots,v_m(m\tau,t)\}\\ &=\left\{\sum_{i=1}^{\tau}\left(\frac{I_i x(t-i)}{\sum_{i=1}^{\tau} I_i}\right),\sum_{i=\tau+1}^{2\tau}\left(\frac{I_i x(t-i)}{\sum_{i=\tau+1}^{2\tau} I_i}\right),\ldots,\sum_{i=(k-1)\tau+1}^{k\tau}\left(\frac{I_i x(t-i)}{\sum_{i=(k-1)\tau+1}^{k\tau} I_i}\right),\ldots,\sum_{i=(m-1)\tau+1}^{m\tau}\left(\frac{I_i x(t-i)}{\sum_{i=(m-1)\tau+1}^{m\tau} I_i}\right)\right\} \end{aligned} \tag{20}$$

Uniform embedding is used when system has one dominant time lag. Nonuniform embedding equation (18) is always the first option since strict periodicity with single dominant recurrence hardly exists in the real-world system. In some cases, when nonuniform embedding equation (18) is not applicable, uniform embedding equation (20) may be used as an alternative. Apparently, if mutual information holds strict periodicity, meaning $\tau_1 \approx (\tau_2-\tau_1) \approx (\tau_3-\tau_2) \approx \cdots \approx (\tau_m-\tau_{m-1})$, then the results from uniform embedding and nonuniform embedding are similar or even identical.

10. Identify Optimal Embedding Dimension

The other embedding parameter that needs to be determined is embedding dimension m in equations (18) and (20). The criterion for obtaining an optimal embedding dimension in a modeling context should be based on optimal model performance, i.e., model input reconstructed from an optimal embedding dimension should lead to optimal model performance. This criterion was applied to find the optimal embedding dimension in our approach. The procedure is as the following.

Starting with embedding dimension $m=1$, we trained the first ANN model (model 1) using input data reconstructed by equation (18) for nonuniform embedding or equation (20) for uniform embedding, and evaluated model output using two model performance statistics, the root-mean-square error (*RMSE*) and coefficient of determination (R^2). Then we increased the embedding dimension m by 1, i.e., $m=2$, and reconstructed new input data using equation (18) or equation (20). With the new input data set, we trained the second ANN model (model 2) and evaluated model output using the same model performance statistics. We found that model 2 performed better than model 1. We repeated this procedure until improvement of model performance became minimal with further increases in m. For example, if model 5 has minimal improvement compared to model 4, then, $m=4$ is the optimal embedding dimension and model 4 is the final model (see an example given in section 11).

10.1002/2014WR016875

11. Simulation of Pore Water Salinity of the Tidal Floodplain at the Loxahatchee River

For the nine probe locations where mutual information clearly showed quasi-periodicity identified in section 7, nonuniform embedding equation (18) was applied and the time lags corresponding to local minimum mutual information points, i.e., $\{\tau_1, \tau_2, \ldots, \tau_m\}$, were used as time lag vector for reconstructing model input. For the rest three probe locations where little quasi-periodicity were shown in their mutual information, we had a difficulty to identify the time lags corresponding to the minimum mutual information. Uniform embedding was applied in this situation using equation (20). The time lag corresponding to the first local minimum mutual information point was taken as system dominant embedding time lag, i.e., $\tau = \tau_1$, resulting in time lag vector $\{\tau, 2\tau, \ldots, m\tau\}$.

The ANN ensemble techniques were used in our simulation since they provide advantages over a single ANN model in potentially improving model simulation ability, increasing model stability, and even overcoming the problem of model over fitting [*Hansen and Salamon*, 1990; *Sharkey*, 1999]. An ensemble (a set of ANNs) was created by randomly initializing ANNs parameters and randomly generating data sets for training, testing, and validating by sampling without replacement. Then the simple averaging method was used to combine output of the set of ANNs as the ensemble output [*Shu and Burn*, 2004]. To determine the optimal ensemble size, a total of 36 ANN models were developed. The *RMSE* of ensembles became stable when ensemble size $n > 14$. Therefore, an ANN ensemble with size of 14 was appropriate for the simulation of pore water salinity at T7, consistent with *Shu and Burn* [2004] who suggested that at least 10 ensemble members were needed in ANN modeling.

For each probe location, a MLP was developed using input time series computed by equation (18) (nonuniform embedding) or (20) (uniform embedding) first with $m=1$. Then, 14 models were trained to form an MLP ensemble for the probe location. The *RMSE* and R^2 of the ANN ensemble output were computed to evaluate model performance. After that, another ANN ensemble was developed using input time series data constructed using equation (18) or (20) by increasing m by 1, i.e., $m=2$, and the *RMSE* and R^2 of the second MLP ensemble output were computed. Model performance improved with increasing m, i.e., *RMSE* decreased and R^2 increased. The final model and optimal m were determined by increasing m until the model performance was not significantly improved.

We take probe location T7-2(0.30) as an example. From the mutual information variation with time lag in Figure 3a-1, the first to fifth time lags corresponding to the local minimum mutual information point are $\tau_1=89$, $\tau_2=193$, $\tau_3=297$, $\tau_4=369$, and $\tau_5=455$ days (the figure does not show $\tau_5=455$) and were taken as the time lag vector (i.e., $\{\tau_1, \tau_2, \tau_3, \tau_4, \tau_5\}=\{89,\ 193,\ 297,\ 369, 455\}$). Then starting with $m=1$ up to 5, five MLP ensembles each with size of 14 were developed using time lag vectors as the following: (1) $\{89\}$ for $m=1$; (2) $\{89,\ 193\}$ for $m=2$; (3) $\{89, 193,\ 297\}$ for $m=3$; (4) $\{89, 193,\ 297,\ 369\}$ for $m=4$; (5) $\{89, 193,\ 297,\ 369, 456\}$ for $m=5$. Figure 7 shows the model performance improved (*RMSE* decreased and R^2 increased) by increasing embedding dimension m. Figure 8 is the comparison between observed and simulated data for $m=1 \sim 5$, showing the same improvement with increase of m. From $m=1$ to 2, $m=2$ to 3, and $m=3$ to 4, the model performance significantly improved. The *RMSE* decreased from 0.067 to 0.058, to 0.0467, and to 0.022 while the R^2 increased from 0.649 to 0.739, to 0.831, and to 0.961, respectively. From $m=4$ to 5, the model showed

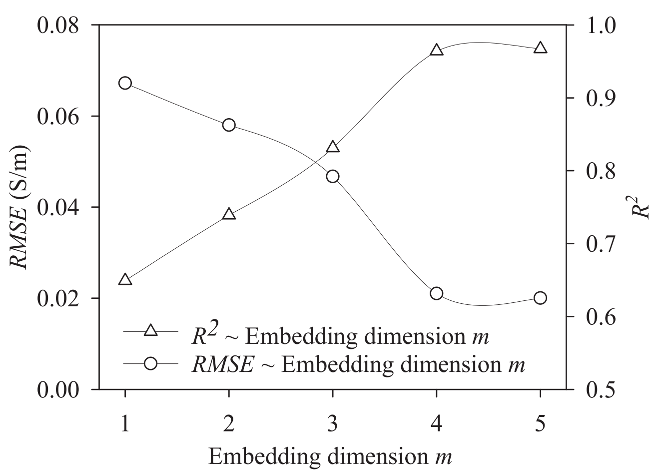

Figure 7. Model performance statistics, *RMSE* and R^2, vary with increasing embedding dimension m in the simulation of electrical conductivity (EC) at probe location T7-2(0.30).

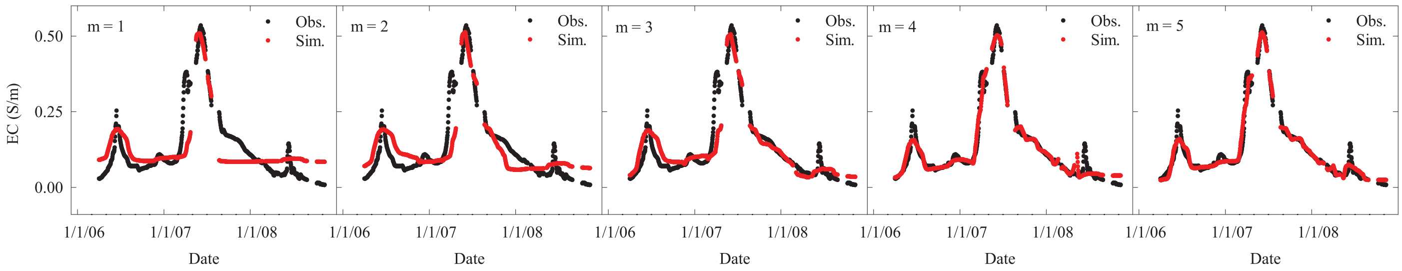

Figure 8. Comparison between observed and simulated electrical conductivity (EC) of probe location T7-2(0.30) showing model output improvements with increases of embedding dimension m from 1 to 5.

insignificant improvement; *RMSE* decreased only from 0.022 to 0.019, and R^2 increased only from 0.961 to 0.967, indicating that reconstructed input data with $m=4$ already contained sufficient information for model output simulation and further increase of m added very little effective information to the reconstructed input. Figure 8 for $m=4$ and 5 visually shows this little improvement. Thus, the optimal embedding dimension $m = 4$ and model 4 can be considered as the final model. The associated embedding window was 369 days.

For the rest of probe locations, it was not necessary to start model development with $m = 1$ but with an m corresponding to an embedding window close to 369 days because they all share the same floodplain system and the embedding windows are supposed be more or less the same as for T7-2(0.30).

Table 1 lists the final embedding time lag vector, embedding method (uniform or nonuniform), embedding dimension, and embedding window for the MLP models developed for 12 probe locations. The time lag vector, embedding method, and embedding dimension differed from location to location. The lag vectors showed some difference in T7-2 and T7-25 though they are relatively close to each other. This is perhaps mostly due to the influence of the heterogeneity of floodplain soils and microtopography on salt accumulation and transport in the floodplain.

Table 2 presents *RMSE* and R^2 of final MLP ensemble performance and Figure 9 compares the ensemble outputs versus observed data for all 12 probe locations. The range (min. ~ max.) of *RMSE* and R^2 of 14 ensemble member models also are provided in Table 2 for model training, testing, and validation. From the table and figure, we see all ensemble outputs had close agreement with the observed data. The simulation results for all 12 probe locations were very good for both individual ensemble member models and ensembles. For ensemble output, the maximum *RMSE* was 0.022 (S/m) at T7-25(0.20) while the minimum R^2 was

Table 2. Performance Statistics, RMSE, and R^2, of Artificial Neural Network Models Developed for 12 Probe Locations at Transect T7 in Loxahatchee River

	RMSE (S/m)	R^2	RMSE (S/m)	R^2	RMSE (S/m)	R^2
	T7-2(0.30)		T7-2(0.14)		T7-2(−0.02)	
Training	0.021–0.024	0.953–0.964	0.010–0.013	0.978–0.982	0.011–0.015	0.971–0.979
Testing	0.020–0.028	0.932–0.970	0.009–0.014	0.977–0.983	0.011–0.015	0.962–0.980
Validation	0.019–0.029	0.936–0.973	0.010–0.014	0.976–0.983	0.011–0.016	0.962–0.980
Ensemble	0.021	0.961	0.011	0.981	0.012	0.978
	T7-25(0.20)		T7-25(−0.03)		T7-25(−0.23)	
Training	0.020–0.029	0.937–0.965	0.020–0.028	0.929–0.959	0.018–0.020	0.959–0.966
Testing	0.020–0.031	0.930–0.968	0.019–0.028	0.902–0.961	0.018–0.021	0.944–0.970
Validation	0.019–0.031	0.920–0.972	0.019–0.031	0.911–0.967	0.018–0.022	0.947–0.969
Ensemble	0.022	0.959	0.021	0.953	0.019	0.962
	T7-90(0.27)		T7-90(0.10)		T7-90(−0.10)	
Training	0.010–0.014	0.963–0.979	0.010–0.011	0.959–0.970	0.009–0.014	0.958–0.980
Testing	0.010–0.014	0.957–0.979	0.010–0.021	0.900–0.973	0.009–0.022	0.914–0.980
Validation	0.010–0.016	0.948–0.978	0.010–0.019	0.900–0.970	0.009–0.018	0.930–0.979
Ensemble	0.010	0.978	0.011	0.961	0.011	0.971
	T7-135(0.37)		T7-135(0.17)		T7-135(−0.10)	
Training	0.013–0.016	0.962–0.973	0.013–0.018	0.930–0.964	0.007–0.010	0.929–0.970
Testing	0.012–0.017	0.956–0.975	0.012–0.015	0.946–0.971	0.006–0.010	0.906–0.978
Validation	0.012–0.017	0.953–0.975	0.012–0.016	0.939–0.967	0.007–0.011	0.897–0.979
Ensemble	0.014	0.970	0.014	0.960	0.008	0.959

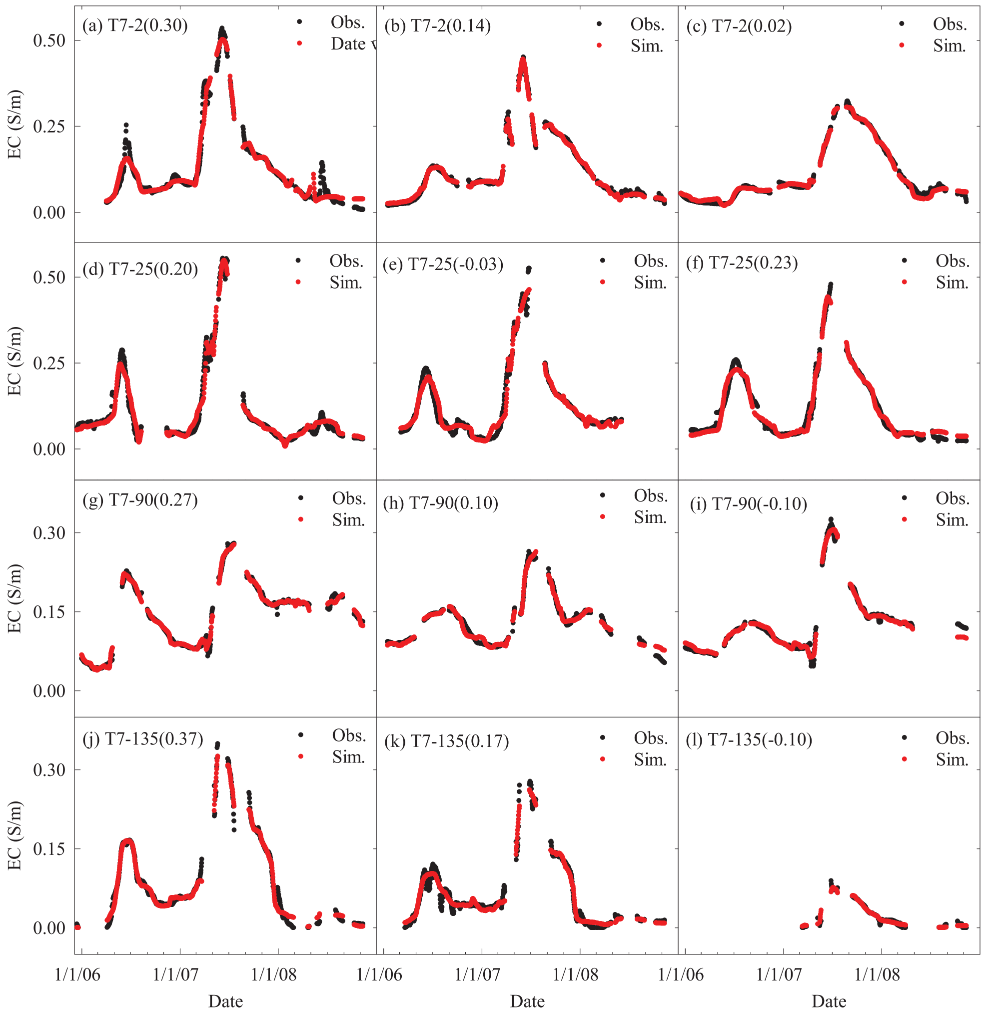

Figure 9. Comparison plots between observed and simulated electrical conductivity (EC) for all 12 probe locations.

0.953 at T7-25(−0.03). For individual ensemble member model performance, the maximum *RMSE* was 0.031 (S/m) at T7-25(−0.03) in validation and at T7-25(0.20) in both testing and validation, while the minimum R^2 was 0.897 at T7-135(−0.10) in validation. This indicates that all the models were well trained and validated and had good simulation ability. The simulation results for three probe locations using uniform embedding method, i.e., T7-25(0.20), T7-25(−0.03), and T7-135(−0.10), showed slightly lower R^2 than the results for nine probe locations using nonuniform embedding. This suggests that nonuniform embedding tends to reconstruct better model input than uniform embedding as a more accurate time lag vector is used in nonuniform embedding.

12. Pore Water Salinity Dynamics and Embedding Window

The variation of river salinity and floodplain pore water salinity exhibits a strong seasonal pattern, following the hydrologic cycle in the Loxahatchee River [*Kaplan et al.*, 2010; *Wan et al.*, 2015], which, we believe, is also related to the embedding window (Table 2). We used Figure 10 to analyze the pore water salinity dynamics. Figure 10a presents 15 min data of river salinity and stage at RK 14.6, and freshwater inflow at Lainhart Dam (LD, Figure 1) from 30 May 2007 to 4 June 2007. Figures 10b and 10c show seasonal pattern of pore water salinity of T7-25 at three depths and river water salinity from 1 May 2005 to 11 November 2008, respectively. Note that during the dry season (prior to 2 June 2007), freshwater inflow from LD was low and river water salinity was high (Figure 10a). High tide pushed brackish water into the floodplain

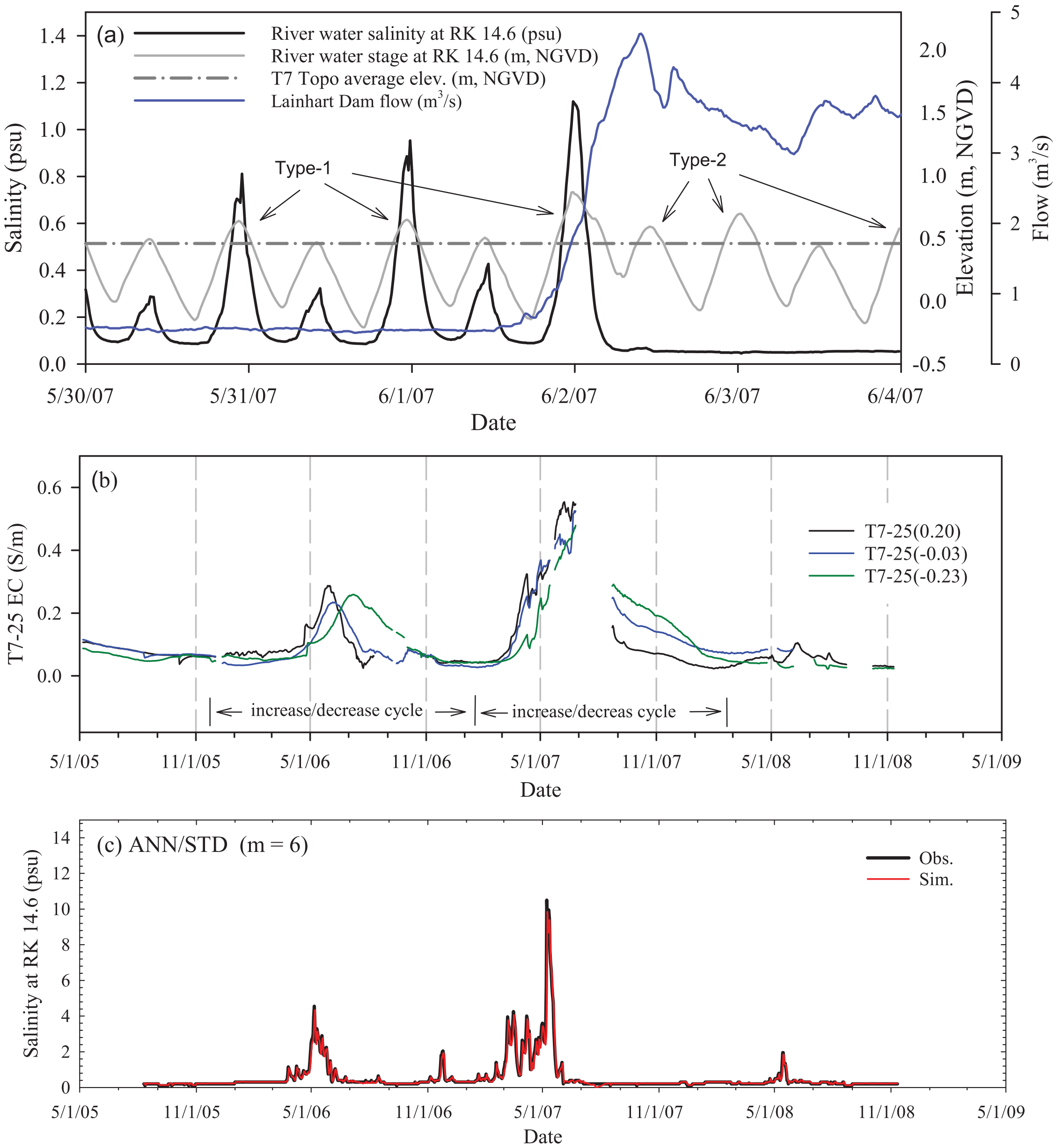

Figure 10. (a) River water salinity and stage at RK 14.6, freshwater inflow over Lainhart Dam from 30 May 2007 to 4 June 2007, and the ground surface topographic elevation of T7; (b) pore water salinity of three probe locations at T7-25 (upper—T7-25(0.20), middle—T7-25(−0.03), and lower—T7-23(−0.23)); (c) observed versus simulated river water salinity at RK 14.6 by ANN using input reconstructed with simple time delay (STD) (embedding dimension $m = 6$).

(type 1 peaks in Figure 10a), resulting in elevated salt levels in pore water salinity. The pore water salinity typically increased at upper probe location, followed by middle then lower probe locations (Figure 10b). In contrast, during the wet season (after 2 June 2007 in Figure 10a), freshwater inflow was high and the river at RK14.6 became nearly fresh. High tide sent freshwater into the floodplain (type 2 peaks in Figure 10a), resulting in decreases in pore water salinity. The salt leaching process takes place in the surface soil first and then in deeper locations (Figure 10b). In addition, reduced rainfall and high evapotranspiration in the dry season may intensify the accumulation of salt in pore water while increased rainfall in the wet season may dilute pore water salinity.

This salt accumulation/depletion cycle in pore water revolves from the dry season to the wet season on an annual basis (Figure 10b). The length and timing of this salt accumulation/depletion oscillation cycle vary from year to year heavily depending on the weather (dry or wet). The maximum length occurs in the driest year (e.g., 2007) and it is around 1 year, which is consistent with the final embedding windows used for

MIWA embedding (Table 2), ranging from 336 to 386 days with an average of 358 days over all the models. Thus, the embedding window seems reflecting the maximum oscillation length of pore water salinity dynamics. The embedding window length is particular important since it determines the amount of information passed from the single scalar time series to the reconstructed vector time series [*Kugiumtzis*, 1996]. If we need to capture the maximum salinity dynamics in simulation, the similar length of data are necessarily in embedding to fully reflect system dynamics. More discussion and analysis on the embedding time window are given in section 13.4.

13. Comparison With Other Modeling Approaches

In this section, we compare ANN using MIWA embedding with other modeling approaches, including multiple linear regression (MLR) model, *k*-nearest neighbors (*k*-NN) model, ANN using partial mutual information (PMI) as input variable selection, and ANN using simple time delay (STD) embedding, to demonstrate the validity of our approach. This exercise was conducted only for T7-2(0.30) as an example. Time lagged river water salinity at RK 14.6 was used as inputs. The river water salinity has some major data gaps (6–29 March 2007 and 7–11 April 2007), which significantly hinder model development for some of the techniques used here (most data in 2007 would be lost due to time lagged). To overcome this issue, we used river salinity data monitored at RK 13.1 (at confluence with Kitching Creek, near Transact 8, Figure 1) to fill the data gaps with a polynomial equation fitted using available data: $y=0.0377x^2-0.048x+0.2327$ with $R^2=0.992$, where x is the salinity at RK13.1 and y is the salinity at RK14.6. Note that these data gaps do not affect the application of MIWA. In equation (17), if there is data missing between τ_{k-1} and τ_k, $v_k(t)$ still can be computed with good quality as long as the number of the missing data is much less than $(\tau_k-\tau_{k-1})$. Also note that no data gap filling was involved in the simulations in section 11. Thus, to certain extent, MIWA could overcome the issue of missing data in modeling.

13.1. Multiple Linear Regression Model

The simple multiple linear regression (MLR) model with k independent variables for simulation of pore water salinity at T7-2(0.30) is given as:

$$y(t)=\beta_0+\beta_1 x(t-1)+\beta_2 x(t-2)+\cdots+\beta_k x(t-k)+\varepsilon \tag{21}$$

where $y(t)$ is pore water salinity at T7-2(0.30), $x(t-i)\ (i=1,\ 2,\ \ldots,\ k)$ are the river water salinity at RK 14.6 at time lag i with respect to $y(t)$, $\beta_i(i=0,\ 1,\ \ldots,\ k)$ are the model parameters, and ε is error assumed independent Gaussian distribution $N(0,\ \sigma^2)$. In order to determine k, R^2 between $y(t)$ and $x(t-i)\ (i=1,\ 2,\ \ldots,\ 300)$ were computed and shown in Figure 11a. When $i=171$, $R^2\approx 0$, therefore $k=171$ was used. The Statistisian, a commercial statistical add-in computer program for Microsoft Excel (http://www.statisticianaddin.com/), was used for the MLR model development. The modeling result (Figures 12a and 12b) has $R^2=0.690$ and $RMSE=0.061\ (\mathrm{S/m})$. This fair performance suggests that a simple linear model is not adequate to quantify the complex relationship between pore water salinity at T7 floodplain and river water salinity at RK 14.6.

13.2. *K*-Nearest Neighbors Model

Farmer and Sidorowich [1987] originally developed the *k*-nearest neighbors (*k*-NN) method for predicting chaotic system through embedding a time series in a state space using delay coordinates. It is a type of instance-based learning, or lazy learning (no training involved), where the prediction of a time series is based on a local approximation using only the nearby observations. The *k*-NN algorithm is among the simplest but powerful of all machine learning algorithms and has been widely used in water resources [*Toth et al.*, 2000; *Solomatine et al.*, 2008; *Elshorbagy et al.*, 2010a, 2010b; *Wu and*

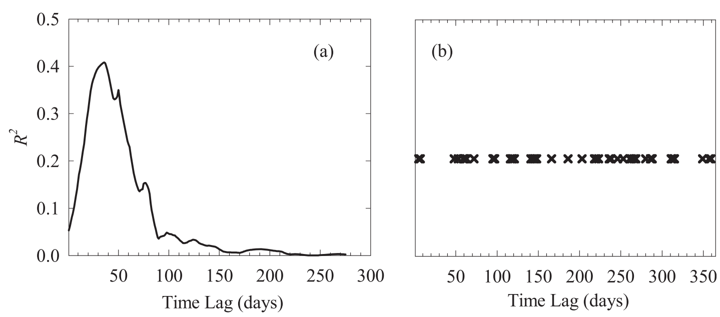

Figure 11. (a) R^2 between river water salinity at RK 14.6 and salinity at T7-2(0.30) versus time lag; (b) time lag of input variable selected using partial mutual information (PMI) algorithm.

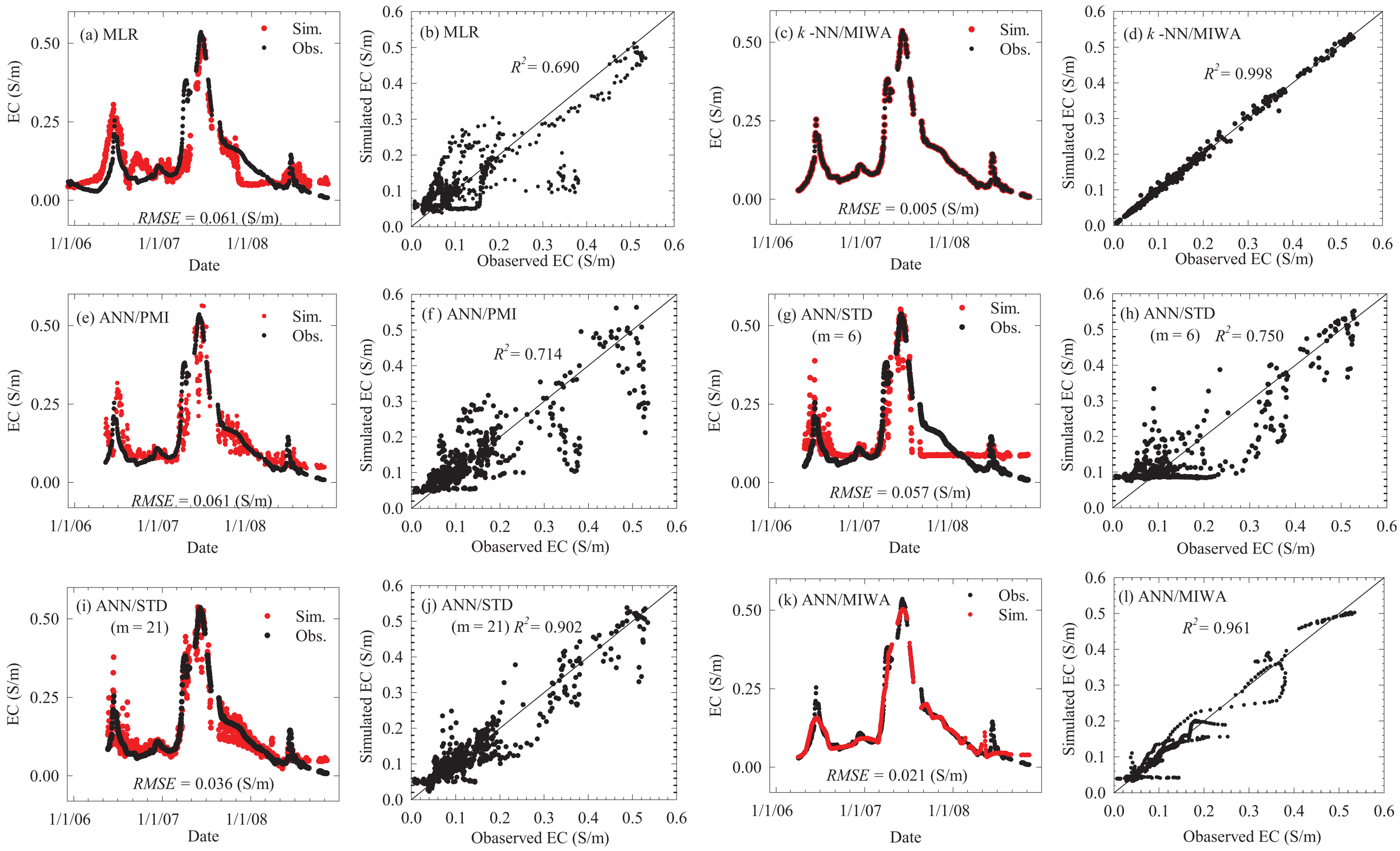

Figure 12. Comparison of observed versus simulated electrical conductivity (EC) of T7-2(0.30) for various models: (a and b) multiple linear regression (MLR); (c and d) *k*-nearest neighbors (*k*-NN) using MIWA for input reconstruction; (e and f) ANN model using partial mutual information (PMI) for input variable selection; (g and h) ANN model using simple time delay (STD) with $m = 6$; (i and j) ANN model using STD with $m = 21$; (k and l) ANN using mutual information weighted average (MIWA).

Chau, 2010]. Once an attractor is correctly unfolded or the phase space of the dynamical system is sufficiently reconstructed, the *k*-NN algorithm has very competitive performance [*Farmer and Sidorowich*, 1987]. In order to verify that the good results achieved in section 11 were not mainly due to the ANN model, we applied the *k*-NN to simulate salinity of T7-2(0.30) (without ANNs involved) using the input reconstructed by MIWA.

Let $\boldsymbol{x}=(x_1,\ x_2,\ \ldots,x_d)$ be a *d*-dimensional feature vector and $y=f(\boldsymbol{x})$. Assume we have *N* pairs of data sample: $(\boldsymbol{x},\ y)^i,\ i=1,\ \ldots,\ N$. If *k*-nearest neighbors to a query point $\boldsymbol{x}_q$ are found from data sample set of size *N*, then prediction of y_q is based on averaging the outcomes of the *k*-nearest neighbors [*Solomatine et al.*, 2008]:

$$y_q=\sum_{j=1}^{k} w_j f(\boldsymbol{x}_j) \Big/ \sum_{j=1}^{k} w_j=\sum_{j=1}^{k} w_j y_j \Big/ \sum_{j=1}^{k} w_j \tag{22}$$

The nearest neighbors are selected based on the distance between $\boldsymbol{x}$ and $\boldsymbol{x}_q$, typically Euclidean distance is used, computed by $D(\boldsymbol{x},\boldsymbol{x}_q)=||\boldsymbol{x}-\boldsymbol{x}_q||=\left(\sum_{i=1}^{d}(x_i-x_{qi})^2\right)^{\frac{1}{2}}$. In equation (22), w_j is the weight for *j*th nearest neighbor, usually a function of the distance with the following options [*Solomatine et al.*, 2008]: (1) linear, $w_j=1-D(\boldsymbol{x_j},\boldsymbol{x}_q)$; (2) inverse, $w_j=1/D(\boldsymbol{x_j},\boldsymbol{x}_q)$; (3) inverse square, $w_j=\left(D(\boldsymbol{x_j},\boldsymbol{x}_q)\right)^{-2}$. One of the drawbacks of *k*-NN is that the predictions are always within the range of historical data (i.e., $y_{\min} \le y_q \le y_{\max}$).

WEKA 3.6.12 [*Bouchkaert et al.*, 2014; *Mark et al.*, 2009] was used to perform *k*-NN simulation. We used the same input reconstructed by MIWA for ANN MLP development of T7-2(0.30). The tenfold cross validation was applied to prevent overfitting. Inverse distance was selected as weights and the optimal number of the nearest neighbors $k=3$, which was obtained by finding the maximum R^2 and minimum *RMSE* through execution of the program. A surprisingly good simulation result with $R^2 = 0.998$ and $RMSE = 0.005$ (S/m) was obtained (Figures 12c and 12d), suggesting that the MIWA embedding played significant role in achieving the good results in section 11.

13.3. ANN Model With Partial Mutual Information Algorithm

To demonstrate the necessity of embedding in our approach, we applied partial mutual information (PMI) variable selection algorithm [*Sharma*, 2000] to select inputs from river water salinity for ANN model to simulate pore water salinity of T7-2(0.30). The PMI algorithm has been demonstrated to be able to properly select optimal input variables for ANN from a set containing complete input candidates [*Sharma*, 2000; *Sharma et al*., 2000; *Bowden et al*., 2005a, 2005b; *May et al*., 2008]. The input candidates are the river water salinity at RK 14.6 from time lag 1 up 369, the same data points as used in MIWA. The assumption by using PMI here is these data points are all potential input candidates required for ANN simulation of the pore water salinity.

The PMI algorithm was proposed by *Sharma* [2000] and extended to partial information (PI) by *Sharma and Mehrotra* [2014]. Computation of PMI is similar to that of MI, where both require estimation of probability densities. *Sharma* [2000] proposed to use an upper limit (95th percentile) of the PMI score as the significance measure of PMI criteria used to select additional input, in which bootstrapping was applied to obtain the 95th percentile confidence limit. A program developed in *May et al*. [2008] was used in this study. With the PMI program, a total of 46 inputs were selected. The time lags corresponding to the selected inputs of the river water salinity are presented in Figure 11b. With this input set, an ensemble of ANN MLP was developed with $R^2=0.714$ and $RMSE=0.061$ (S/m) (Figures 12e and 12f). This fair result implies that the time lagged river water salinity does not contain complete input candidates. A simple combination of the river water salinity is not able to provide sufficient information for ANN model and an appropriate embedding is necessary.

13.4. ANN Model With Simple Time Delay Embedding

If embedding is needed, one may still ask whether MIWA embedding would perform better than simple time delay (STD) embedding in equation (4), which have been used to reconstruct input for ANN and other data-drive models [*Lall et al*., 1996; *Sivakumar et al*., 2002; *Elshorbagy et al*., 2002; *Sivakumar*, 2003, 2007]. The nearest integer above the correlation dimension value provides the minimum dimension of the phase space essential to embed the attractor [*Fraedrich*, 1986]. The correlation dimension of 5.34 obtained from river water salinity in section 8.1 using system time lag of 19 days suggests that the river water salinity most likely be represented as a dynamical system with at least 6 degree of freedom (six differential equations). Thus, input for ANNs was reconstructed with STD by applying equation (4) with $\tau=19$ and $m=6$. The model performance using this input was fair with $R^2=0.75$ and $RMSE=0.057$ S/m (Figures 12g and 12h). We conducted further modeling test by increasing m to reconstructing input. When $m=21$, an improved model performance of $R^2=0.902$ and $RMSE=0.036$ was reached (Figures 12i and 12j). This is still not as good as that with MIWA embedding (Figures 12k and 12l), indicating that MIWA embedding is superior to STD embedding in input reconstruction for ANN.

The difference in above model performance is associated with the amount of information transferred to the reconstructed vector time series from the single scalar time series used for embedding. Generally, observation time series $x(t)$ and $y(t)$ are the projection of system orbits to axes x and y. *Kugiumtzis* [1996] indicated that the length of embedding time window $T_w \geq T_p$, where T_p is the mean orbital period, which is the mean time between two consecutive visits to a local neighborhood and can be approximated from the length of oscillations of observed time series. Let T_x and T_y denote the length of the oscillations in $x(t)$ and $y(t)$. In real-world data, uniform oscillation length usually does not exist in one time series and the maximum is used. Thus, $T_w=(m-1)\tau \geq T_x$ when $x(t)$ is used for embedding and $T_w=(m-1)\tau \geq T_y$ for embedding $y(t)$ (note both m and τ are not necessary to have the same values in two equations). For simulation of $x(t+1)$ by embedding $x(t)$, $T_w \geq T_x$ may usually be enough to capture the oscillation of the time series. However, to ensure that sufficient system memory is retained through embedding $x(t)$ for simulation of $y(t+1)$, the embedding time window should be equal or greater than the maximum length of oscillations in time series $y(t)$ rather than in $x(t)$ when $T_x < T_y$.

For river water salinity, which is dynamically influenced by diurnal/tidal cycles, storm events, and seasonal freshwater under varying time scales, its oscillation is more frequent and shorter (Figure 10c) than that of pore water salinity (Figure 10b). When $m=6$ with $\tau=19$ days was used to embedding input for ANN, the embedding time window $T_w=(6-1)\times 19=115$ days. This embedding window is long enough to capture the oscillations of dynamics in river water salinity, which should result in an effective simulation of river water salinity. For verification purpose, we applied this input to simulate river water salinity and very good model performance was achieved ($R^2=0.939$, Figure 10c). However, the length of oscillation in pore water salinity is much longer, with maximum up to around one year as indicated in Figure 10b. This is why when m increased up to 21, $T_w=(21-1)\times 19=380$ days, a much better result was obtained. Yet MIWA embedding still performed better than STD embedding with $m = 21$.

14. Discussion and Summary

Motivated by practical application, we present an approach to reconstruct input for ANNs using the concept of embedding theory with mutual information. A mutual information weighted average (MIWA) embedding method is developed to compute reconstructed time series components while an optimal embedding dimension is determined by optimizing model performance. The time lags corresponding to the local minimum mutual information points are used to form time lag vector. Our approach is developed purely from the needs of system modeling input reconstruction associated with optimal model performance. The approach allows the reconstructed model input retain maximum information of observed data and a simplistic ANN model structure with minimal input nodes. The algorithm is illustrated for both uniform and nonuniform embedding. The proposed approach provides an efficient and simple way to reconstruct ANN model input for nonlinear system modeling.

The approach was applied to simulate the pore water salinity along a transect in the tidal floodplain of the Loxahatchee River, Florida, knowingly influenced by saltwater intrusion. The model input was reconstructed using river water salinity adjacent to the transect. The results showed that the ANN model developed based on the input reconstructed by the MIWA embedding method closely simulated the soil pore water salinity dynamics along the selected tidal floodplain transect. The ANN model performance using MIWA was compared with a few related modeling techniques, including the time lagged multiple linear regression model, k-nearest neighbors model with MIWA embedding, ANN using partial mutual information for input selection, and ANN using simple time delay embedding. The comparisons demonstrated that MIWA embedding effectively reconstructs unobserved information into model input using only one state variable data for ANN model development.

Due to the presence of noise in observed time series data, mutual information may show weak or no quasi-periodicity with time lag. This causes blurred local minimum mutual information points and their identification becomes difficult and sometimes even subjective. Thus, good quality data and final mutual information filtering are crucial in implementing MIWA embedding. In addition, application of filter on the mutual information more than once can greatly smooth out the noise and remove the subjectivity in identification of local minimum mutual information. The observed data must have sufficient quality to allow for identification of the first local minimum mutual information point to implement uniform embedding when nonuniform embedding is not applicable. The approach appears to be an efficient candidate for reconstruction of ANN model input in the systems where at least one chaotic state variable data is available. We encourage interested researchers to test the method in different real-world systems.

Acknowledgments
The pore water salinity data used in this study are available in UF-HydroBase at University of Florida (http://hb40.ifas.ufl.edu/Public/Logon.aspx) and the river water salinity/stage/flow data are available in the DBHYDRO database of the South Florida Water Management District (SFWMD) (http://www.sfwmd.gov/dhhydro). We would like to thank Michael Chimney, Peter Doering, and Calvin Neidrauer of the SFWMD, as well as the Associate Editor and three anonymous reviewers for their thoughtful reviews and insightful comments that greatly improved the manuscript. The authors would also like to thank Jonathan Dickinson State Park and University of Florida for assistance in data collection, processing, and technique support.

References

Albano, A. M., A. I. Mees, G. C. deGuzman, and P. E. Rapp (1987), Data requirements for reliable estimation of correlation dimensions, in *Chaos in Biological Systems*, edited by H. Degn, A. V. Holden, and L. F. Olsen, *NATO ASI Ser., 138*, 207–220.

Albano, A. M., J. Muench, C. Schwartz, A. I. Mees, and P. E. Rapp (1988), Singular-value decomposition and the Grassberger-Procaccia algorithm, *Phys. Rev. A Gen. Phys.*, *38*(6), 3017–3026.

Barlow, P., and E. G. Reichard (2010), Saltwater intrusion in coastal regions of North America, *Hydrogeol. J.*, *18*, 247–260.

Bishop, C. M. (1995), *Neural Networks for Pattern Recognition*, Oxford Univ. Press, N. Y.

Bouchkaert, R. R., E. Frank, M. Hall, R. Kirkby, P. Reutemann, A. Seewald, and D. Scuse (2014), *WEKA Manual for Version 3-6-12*, Univ. of Waikato, Hamilton, New Zealand.

Bowden, G. J., G. C. Dandy, and H. R. Maier (2005a), Input determination for neural network models in water resources applications, Part 1—Background and methodology, *J. Hydrol.*, *301*, 75–92.

Bowden, G. J., G. C. Dandy, and H. R. Maier (2005b), Input determination for neural network models in water resources applications, Part 2. Case study: Forecasting salinity in a river, *J. Hydrol.*, *301*, 93–107.

Buyuktas, D., and W. W. Wallender (2002), Enhanced subsurface irrigation hydrology model, *J. Irrig. Drain. Eng.*, *128*, 168–174.

Cao, L. (1997), Practical method for determining the minimum embedding dimension of a scalar time series, *Physica D*, *110*(1-2), 43–50.

Casdagli, M., S. Eubank, J. D. Farmer, and J. Gibson (1991), State space reconstruction in the presence of noise, *Physica D*, *51*, 52–98.

Cellucci, C. J., A. M. Albano, and P. E. Rapp (2005), Statistical validation of mutual information calculations: Comparison of alternative numerical algorithms, *Phys. Rev. E*, *71*, 066208-1–066208-14.

Clemente, R. S., S. O. Prasher, R. B. Bonnell, C. A. Madramootoo, and I. N. Mathur (1997), Application of PESTFADE to simulation salt movement in soils, *Can. Water Res. J.*, *22*(2), 167–184, doi:10.4296/cwrj2202167.

Cover, T. M., and J. A. Thomas (1991), *Elements of Information Theory*, John Wiley, N. Y.

Dawson, C. W., and R. L. Wilby (2001), Hydrological modeling using artificial neural networks, *Prog. Phys. Geogr.*, *25*(1), 80–108.

Elshorbagy, A., S. P. Simonovic, and U. S. Panu (2002), Estimation of missing streamflow data using principles of chaos theory, *J. Hydrol*, *255*, 123–133.

Elshorbagy, A., G. Corzo, S. Srinivasulu, and D. P. Solomatine (2010a), Experimental investigation of the predictive capabilities of data driven modeling techniques in hydrology—Part 1: Concepts and methodology, *Hydrol. Earth Syst. Sci.*, *14*, 1931–1941.

Elshorbagy, A., G. Corzo, S. Srinivasulu, and D. P. Solomatine (2010b), Experimental investigation of the predictive capabilities of data driven modeling techniques in hydrology—Part 2: Application, *Hydrol. Earth Syst. Sci.*, *14*, 1943–1961.

Farmer, J. B., and J. J. Sidorowich (1987), Predicting chaotic time series, *Phys. Rev. Lett.*, *59*(4), 845–848.

Fraedrich, K. (1986), Estimating the dimensions of weather and climate attractors, *J. Atmos. Sci.*, *43*(5), 419–132.

Fraser, A. M. (1989), Reconstructing attractors from scalar time series: A comparison for singular system and redundancy criteria, *Physica D*, *34*, 391–404.

Fraser, A. M., and H. L. Swinney (1986), Independent coordinates for strange attractors form mutual information, *Phys. Rev. A Gen. Phys.*, *33*(2), 1134–1140.

Grassberger, P., and I. Procaccia (1983), Measuring the strangeness of strange attractors, *Physica D*, *9*, 189–208.

Hansen, L., and P. Salamon (1990), Neural network ensembles, *IEEE Trans. Pattern Anal. Mach. Intel.*, *12*, 993–1001.

Heath, R. A. (2000), *Nonlinear Dynamics: Techniques and Applications in Psychology*, Lawrence Erlbaum Assoc. Inc., Mahwah, N. J.

Hegger, R., and H. Kantz (1998), Practical implementation of nonlinear time series methods: The TISEAN package, *Chaos*, *9*(2), 413–435.

Hertz, J., A. Krogh, and R. G. Palmer (1991), *Introduction to the Theory of Neural Computation*, Addison-Wesley, Redwood, Calif.

Huang, L. D., and S. Foo (2002), Neural network modeling of salinity variation in Apalachicola River, *Water Res.*, *36*, 356–362.

Judd, K., and A. Mees (1998), Embedding as a modeling problem, *Physica D*, *120*, 273–286.

Kantz, H. (1994), A robust method to estimate the maximal Lyapunov exponent of a time series, *Phys. Lett. A*, *185*(1), 77–87.

Kantz, H., and T. Schreiber (1997), *Nonlinear Time Series Analysis*, Cambridge Univ. Press, Cambridge, U. K.

Kaplan, D., R. Muñoz-Carpena, Y. Wan, M. Hedgepeth, F. Zheng, R. Roberts, and R. Rossmanith (2010), Linking River, floodplain, and vadose zone hydrology to improve restoration of a coastal river affected by saltwater intrusion, *J. Environ. Qual.*, *39*, 1570–1584.

Kennel, M. B., R. Brown, and H. D. Abarbanel (1992), Determining embedding dimension for phase-space reconstruction using geometrical construction, *Phys. Rev. A*, *45*(6), 3403–4311.

Končar, N. (1997), Optimisation methodologies for direct inverse neurocontrol, PhD thesis, Dep. of Comput., Imp. Coll. of Sci., Technol. and Medicine, Univ. of London, London.

Konishi, K., Y. Shirao, H. Kawabata, Y. Inagaki, and Y. Takeda (1993), Fractal dimensions of coupled chaotic systems, *Bull. Univ. Osaka Prefecture, Ser. A*, *42*(1), 27–35.

Kugiumtzis, D. (1996), State space reconstruction parameters in the analysis of chaotic time series—The role of the time window length, *Physica D*, *95*(1), 13–28.

Lall, U., T. Sangoyomi, and H. D. I. Abarbanel (1996), Nonlinear dynamics of the Great Salt Lake: Nonparametric short-term forecasting, *Water Resour. Res.*, *32*(4), 975–985.

Lorenz, E. N. (1963), Deterministic nonperiodic flow, *J. Atmos. Sci. 20*(2), 130–141.

Maier, H. R., and G. C. Dandy (1997), Determining inputs for neural network models of multivariate time series, *Microcomput. Civ. Eng.*, *12*, 353–368.

Maier, H. R., and G. C. Dandy (2000), Neural networks for the prediction and forecasting of water resources variables: A review of modelling issues and applications, *Environ. Modell. Software*, *15*, 101–124.

Maier, H. R., G. C. Dandy, and M. D. Burch (1998), Use of artificial neural networks for modelling cyanobacteria Anabaena spp. in the River Murray, South Australia, *Ecol. Model.*, *105*, 257–272.

Maier, H. R., A. Jain, G. C. Dandy, and K. P. Sudheer (2010), Methods used for the development of neural networks for the prediction of water resource variables in river systems: Current status and future directions, *Environ. Modell. Software*, *25*, 891–909.

Mark, H., E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten (2009), The WEKA data mining software: An update, *ACM SIGKDD Explorations*, *11*(1), 10–18.

Martinerie, J. M., A. M. Albano, A. I. Mees, and P. E. Rapp (1992), Mutual information, strange attractors, and the optimal estimation of dimension, *Phys. Rev. A*, *45*(10), 7058–7064.

May, R. J., H. R. Maier, G. C. Dandy, and T. M. K. G. Fernando (2008), Non-linear variable selection of artificial neural networks using partial mutual information, *Environ. Modell. Software*, *23*, 1312–1326.

May, R. J., G. Dandy, and H. Maier (2011), Review of input variable selection methods for artificial neural networks, in *Artificial Neural Networks—Methodological Advances and Biomedical Applications*, edited by K. Suzuki, pp. 19–44, InTech, Rijeka, Croatia.

Moon, Y.–I., B. Rajagopalan, and U. Lall (1995), Estimation of mutual information using kernel density estimators, *Phys. Rev. E*, *52*(3), 2318–2321.

Morris, J. T. (1995), The mass balance of salt and water in intertidal sediments: Results from North Inlet, south Carolina, *Estuaries*, *18*, 556–567.

Mortl, A., R. Muñoz-Carpena, D. Kaplan, and Y. Li (2011), Calibration of a combined dielectric probe for soil moisture and porewater salinity measurement in organic and mineral coastal wetland soils, *Geoderma*, *161*, 50–62.

Neubauer, S. (2013), Ecosystem responses of a tidal freshwater marsh experiencing saltwater intrusion and altered hydrology, *Estuaries Coasts*, *36*, 491–507, doi:10.1007/s12237-011-9455-x.

Packard, N. H., J. P. Crutchfield, J. D. Farmer, and R. S. Shaw (1980), Geometry from a Time Series, *Phys. Rev. Lett.*, *45*(9), 712–716.

Patel, R. M., S. O. Prasher, P. K. Goel, and R. Bassi (2002), Soil salinity prediction using artificial neural networks, *J. Am. Water Resour. Assoc.*, *38*(1), 91–100.

Remesan, R., A. Shamin, and D. Han (2008), Model data selection using gamma test for daily solar radiation estimation, *Hydrol. Processes*, *22*, 4301–4309.

Roberts, R. E., R. O. Woodbury, and J. Popenoe (2006), Vascular plants of Jonathan Dickinson State Park, *Fla. Sci.*, *69*, 288–327.

Roberts, R. E., M. Y. Hedgepeth, and T. R. Alexander (2008), Vegetational response to saltwater intrusion along the Northwest Fork of the Loxahatchee River within Jonathan Dickinson State Park, *Fla. Sci.*, *71*(4), 383–397.

Rosenstein, M. T., J. J. Collins, and C. J. De Luca (1994), Reconstruction expansion as a geometry-based framework for choosing proper delay times, *Physica D*, *73*, 82–98.

Rössler, O. E. (1976), An equation for continuous chaos, *Phys. Lett. A*, *57*(5), 397–398.
SFWMD (2006), *Restoration Plan for the Northwest Fork of the Loxahatchee River*, West Palm Beach, Fla.
SFWMD/LRD/FDEP (2012), *Addendum to the Restoration Plan for the Northwest Fork of the Loxahatchee River*, West Palm Beach, Fla.
Shannon, C. E. (1948), The mathematical theory of communication, *Bell Syst. Tech. J.*, *27*, 379–423.
Sharkey, A. J. C. (Ed.) (1999), *Combining Artificial Neural Nets: Ensemble and Modular Multi-Net Systems*, Springer, N. Y.
Sharma, A. (2000), Seasonal to interannual rainfall probabilistic forecasts for improved water supply management: Part 1—A strategy for system predictor identification, *J. Hydrol.*, *239*, 240–248.
Sharma, A., and R. Mehrotra (2014), An information theoretic alternative to model a natural system using observational information alone, *Water Resour. Res.*, *50*, 650–660, doi:10.1002/2013WR013845.
Sharma, A., K. C. Luk, I. Cordery, and U. Lall (2000), Seasonal to interannual rainfall probabilistic forecasts for improved water supply management: Part 2—Predictor identification of quarterly rainfall using ocean-atmosphere information, *J. Hydrol.*, *239*, 232–239.
Shu, C., and D. H. Burn (2004), Artificial neural network ensembles and their application in pooled flood frequency analysis, *Water Resour. Res.*, *40*, W09301, doi:10.1029/2003WR002816.
Simunek, J., K. Huang, and M. Th. Van Genuchten (1995), The SWMS_3D code for simulating water flow and solute transport in three-dimensional variably saturated media, Version 1.0., *Res. Rep. 139*, U.S. Salinity Lab., USDA-ARS, Riverside, Calif.
Sivakumar, B. (2003), Forecasting monthly streamflow dynamics in the western United States: A nonlinear dynamical approach, *Environ. Model. Software 18*, 721–728.
Sivakumar, B. (2007), Nonlinear determinism in river flow: Prediction as a possible indicator, *Earth Surf. Processes Landforms*, *32*, 969–979.
Sivakumar, B., A. W. Jayawardena, and T. M. K. G. Fernando (2002), River flow forecasting: Use of phase-space reconstruction and artificial neural networks approaches, *J. Hydrol.*, *265*, 225–245.
Solomatine, D. P., M. Maskey, and D. L. Shrestha (2008), Instance-based learning compared to other data-driven methods in hydrological forecasting, *Hydrol. Processes*, *22*, 275–187, doi:10.1002/hyp.6592.
Takens, F. (1981), Detecting strange attractors in turbulence, in *Dynamical Systems and Turbulence*, vol.898, edited by D. A. Rand and L. S. Young, pp. 365–381, Springer, Berlin.
Toth, E., A. Brath, and A. Montanari (2000), Comparison of short-term rainfall predication models for real-time flood forecasting, *J. Hydrol.*, *239*, 132–147.
Wan, Y., C. Wan, and M. Hedgepeth (2015), Elucidating multidecadal saltwater intrusion and vegetation dynamics in a coastal floodplain with artificial neural networks and aerial photography, *Ecohydrology*, *8*(2), 309–324, doi:10.1002/eco.1509.
Whitney, H. (1936), Differentiable manifolds, *Ann. Math.*, *37*, 645–680.
Wu, C. L., and K. W. Chau (2010), Data-driven models for monthly streamflow time series prediction, *Eng. App. Artif. Intel.*, *23*, 1350–1367, doi:10.1016/j.engappai.2010.04.003.
Zhao, L., F. E. Hicks, and A. R. Fayek (2012), Applicability of multilayer feed-forward neural networks to model the onset of river breakup, *Cold Reg. Sci. Technol.*, *70*, 32–42.